

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра автоматика та управління в технічних системах
(повна назва кафедри)

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри

(підпис) (ініціали, прізвище)

“ ” _____ 20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 126 «Інформаційні системи та технології»
(код і назва спеціальності)

на тему: _____
Система підбору вакансій на основі історичних даних

Виконав: студент __2__ курсу, групи __ІА-73мп__
(шифр групи)

_____ Боярин Олесь Володимирович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н., доц., Дорогий Я.Ю. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2018 року

**Національний технічний університет України
“Київський політехнічний інститут
імені Ігоря Сікорського”**

Факультет інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматики та управління в технічних системах
(повна назва)

Ступінь вищої освіти – другий (магістерський)
(код, назва)

Спеціальність 126 «Інформаційні системи та технології»
(код, назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) О. І. РОЛІК
(ініціали, прізвище)

“ ____ ” _____ 2018_р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Боярину Олесю Володимировичу

(прізвище, ім'я, по батькові)

1. **Тема дисертації** _____ Система підбору вакансій на основі історичних даних _____

Науковий керівник дисертації доцент, к.т.н., доц., Дорогий Ярослав Юрійович _____

затверджені наказом по університету від “ 29 ” жовтня 2018_р. № _____

2. Строк подання студентом дисертації “ 4 ” грудня 2018_р.

3. Об'єкт дослідження: підбір вакансій на основі історичних даних

4. Зміст пояснювальної записки: а) огляд існуючих рішень; б) математична модель;
в) технологічні та архітектурні аспекти системи; г) дослідження архітектури
математичної моделі; д) розробка стартап-проекту.

5. Перелік графічного (ілюстративного) матеріалу

Структурна схема нейронної мережі LSTM + Attention Mechanism, структурна схема нейронної мережі Latent Semantic Model, діаграма послідовності розміщення вакансій, діаграма послідовності пошуку вакансій, діаграма послідовності навчання моделей, діаграма сценаріїв використання, структурна схема системи, діаграма розгортання.

6. Консультанти розділів проекту:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання “ 29 ” жовтня 2018_р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів проекту	Примітка
1	Огляд існуючих рішень та розробка структури системи	10.11.2018	
2	Вибір окремих вузлів	15.11.2018	
3	Розробка структурної схем_системи	20.11.2018	
4	Розробка математичної моделі	25.11.2018	
5	Дослідження та порівняльний аналіз розроблених математичних моделей	29.11.2018	
6	Оформлення текстової та графічної документації	2.12.2018	
7	Представлення до захисту	4.12.2018	

Студент
(підпис) _____
(ініціали, прізвище)

Боярин О.В.

Керівник проекту
(підпис) _____
(ініціали, прізвище)

Дорогий Я.Ю.

РЕФЕРАТ

Магістерська дисертація 107 с., 43 рис., 23 табл., 8 додатки, 25 джерел.

Тема магістерської дисертації “Система підбору вакансій на основі історичних даних”

Актуальність магістерської дисертації обумовлена тим, що алгоритми підбору контенту мають широке застосування у наш час, як наприклад алгоритми підбору вакансій. Дані алгоритми дозволяють підвищити якість ранжування за рахунок моделей, які навчаються на історичних даних, і, як наслідок, покращити якість сервісу. Це, в свою чергу, збільшує конверсію, що свідчить про актуальність даних методів у будь-який час

Об’єктом дослідження є підбір вакансій на основі історичних даних.

Предметом дослідження є точність методів підбору вакансій на основі історичних даних.

Метою цієї роботи є підвищення якості підбору вакансій на основі методів машинного навчання.

Задачею роботи є:

1. дослідити існуючі алгоритми підбору;
2. визначити переваги та недоліки кожного з них для підбору вакансій;
3. на основі отриманих результатів представити отримані результати у вигляді стартап проекту;
4. удосконалити існуючі методи;
5. розробити систему підбору вакансій на основі історичних даних у вигляді веб сервісу;
6. дослідити ефективність удосконаленого методу у порівнянні з існуючими.

Ключові слова: нейронні мережі, система підбору вакансій, машинне навчання, історичні дані.

ABSTRACT

Master's thesis: 107 p., 43 figures, 23 tables, 8 appendixes, 15 sources.

Theme of the master's dissertation "The system of selection of vacancies on the basis of historical data"

The urgency of the master's dissertation is due to the fact that the algorithms of content selection are widely used in our time, such as algorithms for selecting vacancies.

These algorithms allow you to improve the quality of the ranking by the models that are learning on historical data and, as a consequence, improve the quality of the service. This, in turn, increases the conversion, which indicates the relevance of these methods at any time

The object of the study is the selection of vacancies based on historical data.

The subject of the study is the accuracy of the methods for selecting vacancies based on historical data.

The purpose of this work is to improve the quality of job placement based on machine learning techniques.

The task of the work is:

1. examine the existing selection algorithms;
2. identify the advantages and disadvantages of each of them for the selection of vacancies;
3. based on the results obtained, present the results as a startup of the project
4. improve existing methods;
5. develop a system for selecting vacancies based on historical data as a web service;
6. to investigate the efficiency of the improved method in comparison with the existing ones.

Keywords: neural networks, the system of selection of vacancies, machine learning, historical data.

ЗМІСТ

ВСТУП.....	9
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	11
1.1 Огляд існуючих систем пошуку роботи.....	11
1.1.1 Jooble.....	11
1.1.2 LinkedIn.....	13
1.2.3 Indeed.....	14
1.2 Огляд існуючих алгоритмів.....	16
1.2.1 Лінійна регресія.....	16
1.2.2 Логістична регресія.....	22
1.2.4 Древа рішень.....	25
1.2.4 Нейронні мережі.....	28
1.3 Метрики оцінювання.....	29
1.4 Функції активації.....	31
1.4.1 Функція сигмоїди.....	31
1.4.2 Тангенціальна функція.....	31
1.4.3 Softmax.....	32
1.5 Вимоги до системи.....	32
1.5.1 Функціональні вимоги.....	32
1.5.2 Нефункціональні вимоги.....	33
2 МАТЕМАТИЧНА МОДЕЛЬ.....	36
2.1 Постановка задачі.....	36
2.2 Skip-gram.....	37
2.3 LSTM.....	41
2.4 Attention Mechanism.....	45
2.5 Latent Semantic Model.....	47
3 ТЕХНОЛОГІЧНІ ТА АРХІТЕКТУРНІ АСПЕКТИ СИСТЕМИ.....	51
3.1 Архітектура системи.....	51

3.1.1 Мікросервісна архітектура.....	51
3.1.2 Сервіси системи.....	53
3.2 Структурна схема системи.....	56
3.3 Вибір та обґрунтування елементів та технологій.....	57
3.4 Сценарії використання системи.....	63
3.5 Інтерфейс користувача.....	63
3.6 Тестування системи.....	64
4 ДОСЛІДЖЕННЯ АРХІТЕКТУРИ МАТЕМАТИЧНОЇ МОДЕЛІ.....	67
4.1 Методи порівняння алгоритмів.....	67
4.2 Порівняльний аналіз моделей.....	68
4.2.1 Модель логістична регресія.....	68
4.2.2 Модель RandomForest.....	71
4.2.3 Модель GradientBoosting.....	74
4.2.4 LSTM + Attention Mechanism.....	78
4.2.5 LSTM + Attention Mechanism + Latent Semantic Model.....	80
4.3 Оцінка генералізації моделі.....	83
4.3.1 LSTM + Attention Mechanism.....	83
4.3.2 LSTM + Attention Mechanism + Latent Semantic Model.....	84
4.3.3 Latent Semantic Model.....	86
5 РОЗРОБКА СТАРТАП-ПРОЕКТУ.....	88
5.1 Опис ідеї проекту.....	88
5.2 Технологічний аудит ідеї проекту.....	90
5.3 Аналіз ринкових можливостей запуску.....	91
5.4 Розроблення ринкової стратегії проекту.....	99
5.5 Розроблення маркетингової програми.....	102
ВИСНОВКИ.....	107
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	109
ДОДАТКИ.....	111

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface

LSTM – Long Short Term Memory

LSM – Latent Semantic Model

CLSL – Convolutional Latent Semantic Model

RMSE – Root Mean Squared Error

REST – Representational State Transfer

HTTP – Hyper Text Transfer Protocol

TCP – Transmission Control Protocol

NDCG – Normalized Discounted Cumulative Gain

SQL – Structured Query Language

СУБД — система управління базами даних

ВСТУП

У нинішній час бізнес починає активно впроваджувати сервіси побудовані за допомогою алгоритмів машинного навчання. Сервіси такого роду дають змогу вирішити задачі, які неможливо вирішити за допомогою чітких, заданих правил. Це дає змогу максимально адаптувати рішення для потреб бізнесу.

Одним із найпопулярніших алгоритмів, який широко застосовується для вирішення задач бізнесу є нейронні мережі. Це обумовлюється тим, що нейронні мережі дають змогу якісно вирішити задачу, для цього потрібні лише дані, які певним чином стосуються контексту задачі. Специфікою їх роботи є те, що вони здатні знаходити нелінійні залежності у вхідних даних. Це дає змогу знайти певні залежності в процесах, які до того були приховані.

Ключовим фактором використання нейронних мереж в бізнесі є підвищення конверсії за рахунок покращення якості роботи сервісів, які раніше вирішувалися за допомогою класичних методів.

До таких сервісів можна віднести сервіси рекомендацій, сервіси підбору клієнтського контенту, прогнозування аномалій тощо.

Спроба розробити систему, яка дозволить якісно підбирати вакансії на основі історичних даних залишає обидві сторони у виграші. З одного боку користувач, який шукає роботу. З іншого — роботодавець, якому необхідно якомога швидше знайти кандидата на вакансію. В даному випадку відстежується пряма залежність між якістю підбору вакансій і конверсією. Відповідно, чим вище якість підбору вакансій, тим більше користувачів знаходять роботу за короткий проміжок часу. З цього випливає, що чим більше відношення кількості користувачів, які знайшли роботу, до загального часу пошуку — тим більша конверсія.

Актуальність магістерської дисертації обумовлена тим, що алгоритми підбору контенту мають широке застосування у наш час, як наприклад алгоритми підбору вакансій. Дані алгоритми дозволяють підвищити якість ранжування за рахунок моделей, які навчаються на історичних даних, і, як наслідок, покращити

якість сервісу. Це, в свою чергу, збільшує конверсію, що свідчить про актуальність даних методів у будь-який час

Об'єктом дослідження є підбір вакансій на основі історичних даних.

Предметом дослідження є точність методів підбору вакансій на основі історичних даних.

Метою цієї роботи є підвищення якості підбору вакансій на основі методів машинного навчання.

Задачею роботи є:

1. дослідити існуючі алгоритми підбору;
2. визначити переваги та недоліки кожного з них для підбору вакансій;
3. на основі отриманих результатів представити отримані результати у вигляді стартап проекту;
4. удосконалити існуючі методи;
5. розробити систему підбору вакансій на основі історичних даних у вигляді веб сервісу;
6. дослідити ефективність удосконаленого методу у порівнянні з існуючими.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Огляд існуючих систем пошуку роботи

1.1.1 Jooble

Jooble - це вертикальна пошукова система, яка агрегує оголошення про вакансії з тисяч джоббордів, газет, кар'єрних сторінок підприємств і рекрутингових агентств. Jooble працює в 66 країнах і підтримує більш ніж 24 мови. Офіс компанії знаходиться в Києві, Україна [1].

Jooble збирає оголошення про вакансії з джоббордів, сторінок компаній, газет, рекрутингових агенств. Пошукач розпочинає пошук на сайті з введення міста і назви вакансії. За допомогою фільтрів можна сортувати оголошення за віддаленістю від міста, датою, зарплатою і назвою підприємства-роботодавця.

Алгоритм пошук оголошень про роботу враховує сотні параметрів і підбирає для користувача релевантні і актуальні вакансії. Jooble пропонує вакансії як з повною зайнятістю, так і з частковою, а також практики і фрілансерські вакансії. При кліку на оголошення користувач перенаправляється на відповідну сторінку компанії, де він зможе знайти контакти роботодавця і подати резюме.

На рисунку 1.1 зображена головна сторінка сайту Jooble, на якій одразу можна ввести пошуковий запит, та опинитися на сторінці пошукових результатів, які зображені на рисунку 1.2.

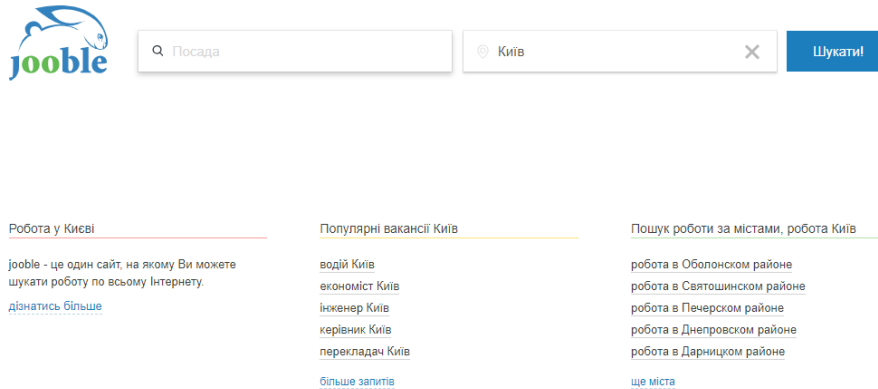


Рисунок 1.1 — Головна сторінка сайту Jooble

Також, опинившись на сторінці пошукових результатів можна обрати додаткові фільтри, як наприклад, дату публікації, зарплату тощо.

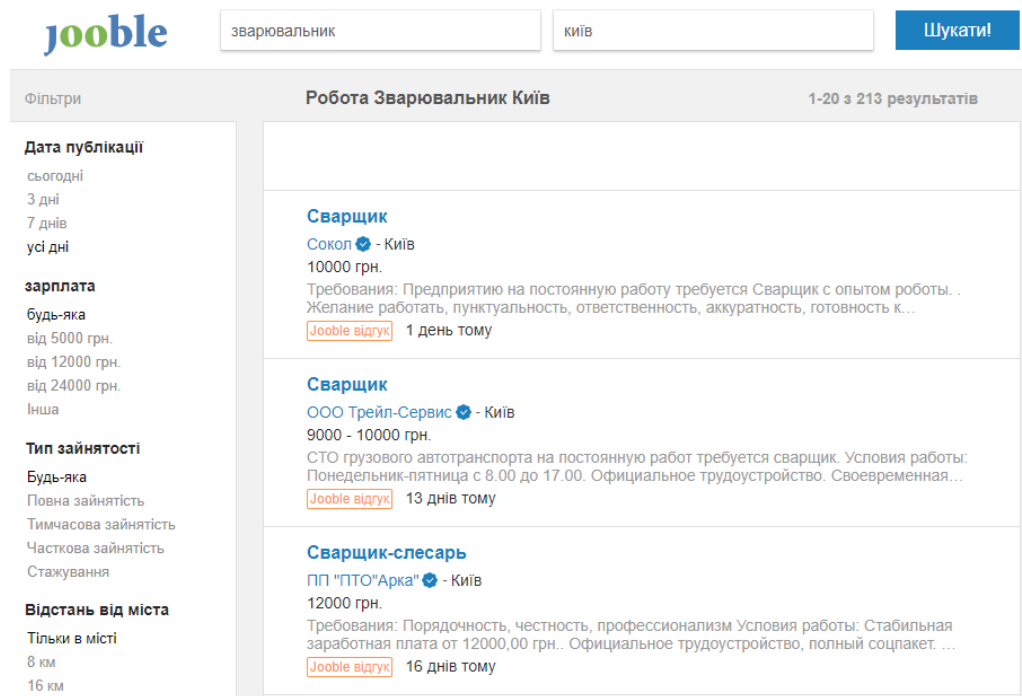


Рисунок 1.2 — Сторінка пошукових результатів на сайті Jooble

1.1.2 LinkedIn

LinkedIn — соціальна мережа для пошуку і встановлення ділових контактів. У LinkedIn зареєстровано понад 400 мільйонів користувачів (на 2015 рік), що представляють 150 галузей бізнесу з 200 країн. Сайт представлений англійською, французькою, німецькою, італійською, португальською, іспанською та російською мовами [2].

LinkedIn надає можливість зареєстрованим користувачам створювати і підтримувати список ділових контактів. Контакти можуть бути запрошені як з сайту, так і ззовні, проте LinkedIn вимагає попереднє знайомство з контактами. У випадку, коли користувач не має прямого зв'язку з контактом, він може бути представленим через інший контакт. Користувачі LinkedIn можуть використовувати список контактів у різних цілях:

- бути представленими через існуючі контакти і розширювати зв'язки;
- здійснювати пошук компаній, людей, груп за інтересами;
- публікувати професійні резюме і здійснювати пошук роботи; Рекомендувати і бути рекомендованими;
- публікувати вакансії;
- створювати групи за інтересами.

Для пошуку на сайті LinkedIn необхідно бути авторизованим в системі, ввівши пошуковий запит, користувач одразу опиняється на сторінці пошукових результатів, як продемонстровано на рисунку 1.3.

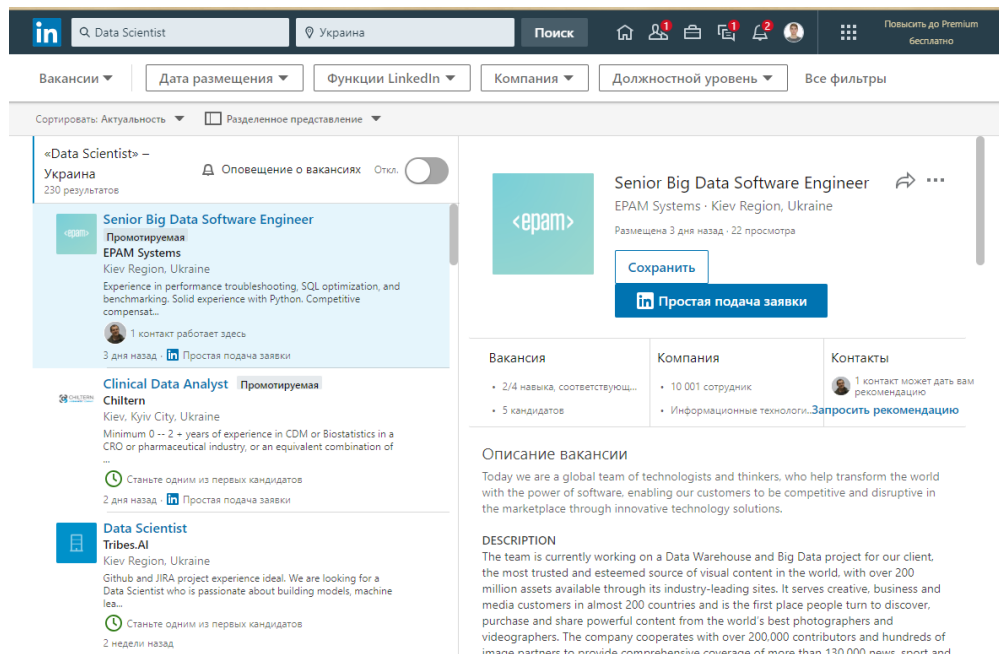


Рисунок 1.3 — Сторінка пошукових результатів сайту LinkedIn

1.2.3 Indeed

Indeed – це американська пошукова система вертикального типу з пошуку роботи по всьому світу, запущена в листопаді 2004 року. Indeed, в даний час доступний в більш ніж 60 країнах і 28-ма мовами [3].

На сайті зібрані списки вакансій з тисяч веб-сайтів, включаючи дошки оголошень про вакансії, кадрові фірми, сайтів асоціацій та компаній. У 2011 році InDeed реалізував функціонал відклику претендента на вакансію безпосередньо на сайті Indeed, а також функції розміщення і зберігання резюме.

Indeed має наступні сервіси:

- пошук роботи;
- рекомендація вакансій;
- тренінги щодо роботи;
- завантаження резюме, зберігання та пошук;

- тенденції галузі;
- пошук зарплат;
- індекс конкуренції;
- форуми веб-сайтів.

На рисунку 1.4 зображено сторінку пошукових результатів сайту Indeed, як видно, на сторінці також можна застосувати різні фільтри для уточнення пошукового запиту.

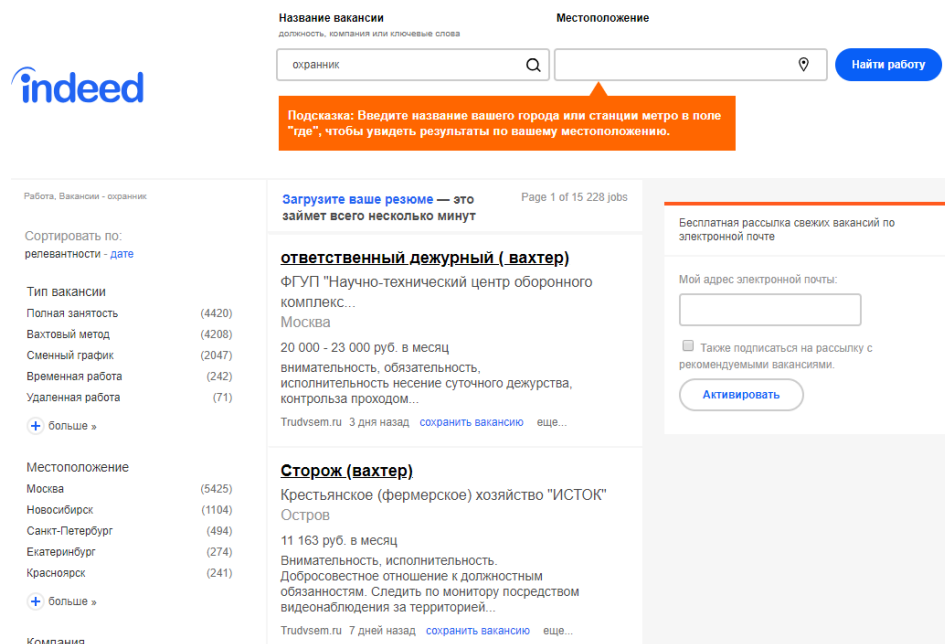


Рисунок 1.4 — Сторінка пошукових результатів на сайті Indeed

1.2 Огляд існуючих алгоритмів

Загалом, методи машинного навчання з вчителем (supervised learning), які розглядаються в даній роботі, працюють та оцінюються наступним чином:

1. На вхід алгоритму подаються вхідні данні, як правило дані складаються з певних факторів, за якими модель знаходить певні закономірності в них для прогнозування кінцевого результату. Пошук закономірностей відбувається за рахунок різних математичних методів оптимізації. Таким чином, модель навчається вирішувати певну задачу.
2. Після навчання моделі на вхід подається тестовий набір даних, який використовується для оцінки якості роботи вказаного алгоритму. На цьому етапі можна оцінити якість прогнозування у порівнянні з іншими алгоритмами.
3. Оцінка генералізації моделі. На цьому етапі проводиться оцінка моделі в реальному світі. Як правило, порівнюються результати навчання та тестування моделі, різниця між якими й відражає якість генералізації. Дана перевірка дозволяє оцінити масштаб перенавчання моделі (overfitting).

1.2.1 Лінійна регресія

Лінійна регресія – одна з регресійних моделей, яка встановлює залежність однієї змінної (цільової) від іншої, або ряду інших змін (факторів, регресорів, незалежних змінних) з лінійною функцією залежності [4].

Регресійна модель:

$$y = f(x, b) + \varepsilon, E(\varepsilon) = 0 \quad (1.2)$$

де b — параметри моделі;

ε — випадкова помилка моделі.

називається лінійною регресією, якщо функція регресії $f(x, b)$ має вигляд:

$$f(x, b) = b_0 + b_1x_1 + b_2x_2 + \dots + b_kx_k \quad (1.2)$$

Коефіцієнти лінійної регресії показують швидкість зміни залежної змінної по даному фактору, при фіксованих інших факторах (в лінійної моделі ця швидкість постійна):

$$\forall j \, b_j = \frac{\partial f}{\partial x_j} = \text{const} \quad (1.3)$$

Параметр b_0 , при якому немає факторів, часто називають константою. Формально - це значення функції при нульовому значенні всіх факторів. Для аналітичних цілей зручно вважати, що константа - це параметр при «факторі» рівному 1 (або іншій довільній сталій, тому константою називають також і цей «фактор»). У такому випадку, якщо перенумерувати фактори і параметри вихідної моделі з урахуванням цього (залишивши позначення загальної кількості чинників - k), то лінійну функцію регресії можна записати як функцію, яка формально не містить константу:

$$f(x, b) = b_1x_1 + b_2x_2 + \dots + b_kx_k = \sum_{j=1}^k b_jx_j = x^T b \quad (1.4)$$

де $x^T = (x_1, x_2, \dots, x_k)$ – вектор регресорів;

$b = (b_1, b_2, \dots, b_k)^T$ – вектор-стовпець параметрів (коефіцієнтів).

Лінійна модель може бути як з константою, так і без константи. Тоді в цьому поданні перший фактор або дорівнює одиниці, або є звичайним фактором.

В окремому випадку, коли фактор єдиний (без урахування константи), говорять про парну або найпростішу лінійну регресію:

$$y_i = a + bx_i + \varepsilon_i \quad (1.5)$$

Коли кількість чинників (без урахування константи) більше 1-го, то говорять про множинну регресію.

Більшість додатків, в яких реалізована лінійна регресія використовують матричне представлення моделі для зручності та збільшення ефективності

обчислювання. Нехай дана вибірка обсягом n спостережень змінних y і x . Нехай, t - номер спостереження у вибірці.

Відповідно $x_t^T = (x_{t1}, x_{t2}, \dots, x_{tk})$ – вектор регресорів в t -му спостереженні. Тоді лінійна регресійна залежність має місце в кожному спостереженні:

$$y_t = b_1 x_{t1} + b_2 x_{t2} + \dots + b_k x_{tk} = \sum_{j=1}^k b_j x_{tj} = x_t^T b + \varepsilon_t, E(\varepsilon_t) = 0, t = 1..n \quad (1.6)$$

На рисунку 1.5 зображено представлено зображення прикладу моделі лінійної регресії, де спостереження (червоні) приймаються як результат випадкових відхилень (зелені) від базового співвідношення (синій) між залежною змінною (y) та незалежною змінною (x).

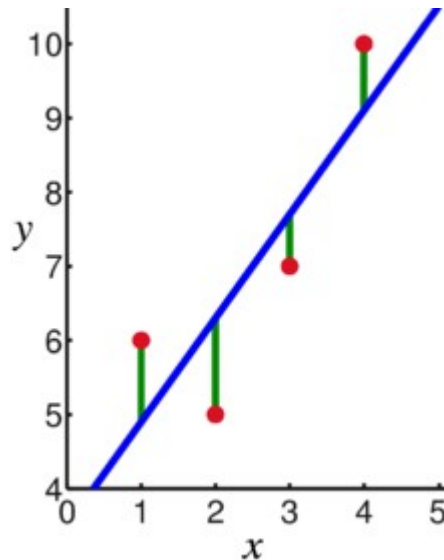


Рисунок 1.5 – Приклад зображення моделі лінійної регресії [4]

Позначивши:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix} \text{ – вектор спостережень залежної змінної } y$$

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1k} \\ x_{21} & x_{22} & \dots & x_{2k} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nk} \end{pmatrix} - \text{матриця факторів}$$

$$\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_n \end{pmatrix} - \text{вектор випадкових помилок}$$

Тоді модель лінійної регресії можна представити в матричній формі:

$$y = Xb + \varepsilon \quad (1.7)$$

Для визначення параметрів моделі, таких як коефіцієнти матриці факторів, використовують метод найменших квадратів.

Метод найменших квадратів - математичний метод, який застосовується для вирішення різних задач, який базується на мінімізації суми квадратів відхилень деяких функцій від шуканих змінних [5]. Він може бути використаний для рішення невизначених систем рівнянь (коли кількість рівнянь перевищує кількість невідомих), для пошуку рішення в звичайних нелінійних системах рівнянь, для апроксимації точкових значень деякої функції. Метод найменших квадратів є одним з базових методів регресійного аналізу для оцінки невідомих параметрів регресійних моделей за вибіркоvim даним.

У випадку із лінійною регресією метод найменших квадратів обчислюється наступним чином:

Нехай регресійна залежність є лінійною:

$$y_t = \sum_{j=1}^k b_j x_{tj} + \varepsilon = x_t^T b + \varepsilon_t \quad (1.8)$$

Нехай y - вектор-стовпець спостережень пояснюється змінною, а X - це $(n \times k)$ матриця спостережень факторів (рядки матриці - вектори значень факторів в даному спостереженні, по стовпцям - вектор значень даного чинника у всіх спостереженнях). Матричне подання лінійної моделі має вигляд:

$$y = Xb + \varepsilon \quad (1.9)$$

Тоді вектор оцінок пояснюваної змінної і вектор залишків регресії дорівнюватимуть:

$$\hat{y} = Xb, \quad e = y - \hat{y} = y - Xb \quad (1.10)$$

відповідно сума квадратів залишків регресії буде дорівнювати:

$$RSS = e^T e = (y - Xb)^T (y - Xb) \quad (1.11)$$

Диференціюючи цю функцію по вектору параметрів b і прирівнявши похідні до нуля, на виході отримується систему рівнянь (в матричній формі):

$$(X^T X)b = X^T y \quad (1.12)$$

У матричній формі ця система рівнянь виглядає наступним чином:

$$\begin{pmatrix} \sum x_{t1}^2 & \sum x_{t1}x_{t2} & \sum x_{t1}x_{t3} & \dots & \sum x_{t1}x_{tk} \\ \sum x_{t2}x_{t1} & \sum x_{t2}^2 & \sum x_{t2}x_{t3} & \dots & \sum x_{t2}x_{tk} \\ \sum x_{t3}x_{t1} & \sum x_{t3}x_{t2} & \sum x_{t3}^2 & \dots & \sum x_{t3}x_{tk} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_{tk}x_{t1} & \sum x_{tk}x_{t2} & \sum x_{tk}x_{t3} & \dots & \sum x_{tk}^2 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_k \end{pmatrix} = \begin{pmatrix} \sum x_{t1}y_t \\ \sum x_{t2}y_t \\ \sum x_{t3}y_t \\ \vdots \\ \sum x_{tk}y_t \end{pmatrix} \quad (1.13)$$

Якщо в модель включена константа, то $x_{t1}=1$ при всіх t , тому в лівому верхньому кутку матриці системи рівнянь знаходиться кількість спостережень n , а в інших елементах першого рядка і першого стовпчика - просто суми значень змінних: і $\sum x_{tj}$ перший елемент правої частини системи – $\sum y_t$.

Вирішення цієї системи рівнянь і дає загальну формулу оцінок методу найменших квадратів для лінійної моделі:

$$\hat{b}_{OLS} = (X^T X)^{-1} X^T y = \left(\frac{1}{n} X^T X \right)^{-1} \frac{1}{n} X^T y = V_x^{-1} C_{xy} \quad (1.14)$$

Для аналітичних цілей видається корисним останнє представлення формули (в системі рівнянь при діленні на n , замість сум фігурують середні арифметичні). Якщо в регресійній моделі дані центровані, то в такому представлення перша матриця представляє собою вибіркове коваріаційної матриці факторів, а друга –

вектор коваріацій факторів із залежною змінною. Окрім того, якщо дані нормовані – то перша матриця представляє собою вибіркове кореляційної матриці факторів, а другий вектор – вектор вибіркових кореляцій факторів із залежною змінною.

Важливою властивістю оцінок матриці найменших квадратів для моделей з константою – лінія побудованої регресії (Рисунок 1.5) проходить через центр важкості вибіркових даних, тобто виконується нерівність:

$$\bar{y} = \hat{b}_1 + \sum_{j=2}^k \hat{b}_j \bar{x}_j \quad (1.15)$$

Наприклад, в крайньому випадку, коли єдиним регресором є константа, отримуємо, що оцінка методу найменших квадратів для єдиного параметру дорівнює середньому значенню пояснювальної змінної.

У разі парної лінійної регресії, $y_t = a + bx_t + \varepsilon_t$ коли оцінюється лінійна залежність однієї змінної від іншої, формули розрахунку спрощуються (можна обійтися без матричної алгебри). Система рівнянь має вигляд:

$$\begin{pmatrix} 1 & \bar{x} \\ \bar{x} & \overline{x^2} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \bar{y} \\ \overline{xy} \end{pmatrix} \quad (1.16)$$

Звідси нескладно знайти оцінки коефіцієнтів:

$$\begin{cases} \hat{b} = \frac{\text{Cov}(x, y)}{\text{Var}(x)} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}, \\ \hat{a} = \bar{y} - b\bar{x}. \end{cases} \quad (1.17)$$

Якщо дані апроксимуються поліноміальною функцією регресії однієї змінної

то, $f(x) = b_0 + \sum_{i=1}^k b_i x^i$ сприймаючи ступені x^i як незалежні фактори для кожного i можна оцінити параметри моделі виходячи із загальної формули оцінки параметрів лінійної моделі. Для цього в загальну формулу достатньо врахувати, що при такій інтерпретації $x_{it}x_{jt} = x_t^i x_t^j = x_t^{i+j}$ і $x_{it}y_t = x_t^i y_t$. Відповідно, матричні рівняння в даному випадку приймуть вигляд:

$$\begin{pmatrix} n & \sum_n x_t & \dots & \sum_n x_t^k \\ \sum_n x_t & \sum_n x_t^2 & \dots & \sum_n x_t^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_n x_t^k & \sum_n x_t^{k+1} & \dots & \sum_n x_t^{2k} \end{pmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix} = \begin{bmatrix} \sum_n y_t \\ \sum_n x_t y_t \\ \vdots \\ \sum_n x_t^k y_t \end{bmatrix}. \quad (1.18)$$

1.2.2 Логістична регресія

Логістична регресія, на відміну від лінійної регресії, використовується для бінарної класифікації, тобто прогнозування ймовірності появи однієї із двох подій.

Логістична регресія застосовується для передбачення ймовірності виникнення деякої події за значеннями множини ознак. Для цього вводиться так звана залежна змінна y , яка приймає лише одне з двох значень - як правило, це число 0 (подія не відбулася) і 1 (подія відбулася), і множини незалежних змінних (також званих ознаками або регресорами) — x_1, x_2, \dots, x_n , на основі значень яких потрібно обчислити ймовірність прийняття того чи іншого значення залежної змінної [6]. Як і в разі лінійної регресії, для простоти запису вводиться фіктивна ознака $x_0 = 1$.

Робиться припущення про те, що ймовірність настання події $y = 1$ дорівнює:

$$P\{y=1 | x\} = f(z),$$

де $z = \theta^T x = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n$ — вектори-стовпці значень незалежних змінних і $1, x_1, \dots, x_n$ параметрів (коефіцієнтів регресії) - дійсних чисел $\theta_0, \dots, \theta_n$;

$f(z)$ — так звана логістична функція (сигмоїда), яка має вигляд:

$$f(z) = \frac{1}{1 + e^{-z}} \quad (1.19)$$

Графік логістичної функції зображений на рисунку 1.6.

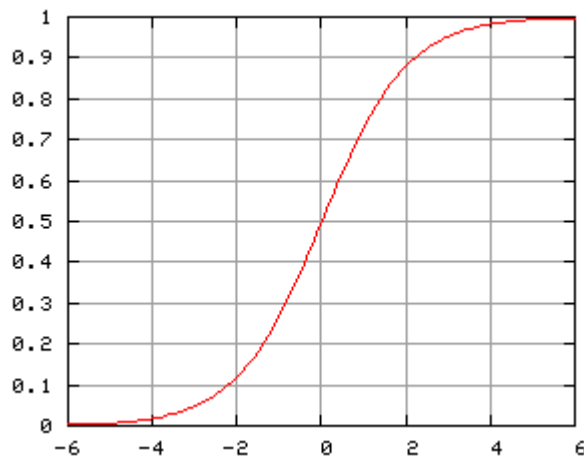


Рисунок 1.6 – Графік логістичної функції [6]

Оскільки y приймає лише значення 0 і 1, то ймовірність першого можливого значення дорівнює:

$$P\{y = 0 | x\} = 1 - f(z) = 1 - f(\theta^T x) \quad (1.20)$$

Функцію розподілу y при заданому x можна записати в такому вигляді:

$$P\{y | x\} = f(\theta^T x)^y (1 - f(\theta^T x))^{1-y}, \quad y \in \{0, 1\}. \quad (1.21)$$

Фактично, це є розподіл Бернуллі з параметром, рівним $f(\theta^T x)$.

Для підбору параметрів $\theta_0, \dots, \theta_n$ необхідно скласти навчальну вибірку, що складається з наборів значень незалежних змінних і відповідних їм значень залежної змінної y . Формально, це множина пар $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$, де $x^{(i)} \in \mathbb{R}^n$ – вектор значень незалежних змінних, а $y^{(i)} \in \{0, 1\}$ – відповідне їм значення y . Кожна така пара називається навчальним прикладом.

Зазвичай використовується метод максимальної правдоподібності, згідно з яким вибираються параметри θ , які максимізують значення функції правдоподібності на навчальній вибірці:

$$\hat{\theta} = \operatorname{argmax}_{\theta} L(\theta) = \operatorname{argmax}_{\theta} \prod_{i=1}^m P\{y = y^{(i)} | x = x^{(i)}\} \quad (1.22)$$

Максимізація функції правдоподібності еквівалентна максимізації її логарифма:

$$\ln L(\theta) = \sum_{i=1}^m \log P\{y = y^{(i)} \mid x = x^{(i)}\} = \sum_{i=1}^m y^{(i)} \ln f(\theta^T x^{(i)}) + (1 - y^{(i)}) \ln(1 - f(\theta^T x^{(i)})) \quad (1.23)$$

де $\theta^T x^{(i)} = \theta_0 + \theta_1 x_1^{(i)} + \dots + \theta_n x_n^{(i)}$.

Для максимізації цієї функції може бути застосований, наприклад, метод градієнтного спуску. Він полягає у виконанні наступних ітерацій, починаючи з деякого початкового значення параметрів θ :

$$\theta := \theta + \alpha \nabla \ln L(\theta) = \theta + \alpha \sum_{i=1}^m (y^{(i)} - f(\theta^T x^{(i)})) x^{(i)}, \alpha > 0 \quad (1.24)$$

Для поліпшення узагальнюючої здатності отриманої моделі, тобто зменшення ефекту перенавчання, на практиці часто розглядається логістична регресія з регуляризації.

Регуляризація полягає в тому, що вектор параметрів θ розглядається як випадковий вектор з деякою заданою апіорної щільністю розподілу $p(\theta)$. Для навчання моделі замість методу найбільшої правдоподібності використовується метод максимізації апіорної оцінки, тобто знаходяться параметри θ , які максимізують величину:

$$\prod_{i=1}^m P\{y^{(i)} \mid x^{(i)}, \theta\} \cdot p(\theta). \quad (1.25)$$

В якості апіорного розподілу часто виступає багатовимірний нормальний розподіл $N(0, \sigma^2 I)$ з нульовим середнім і матрицею коваріації $\sigma^2 I$ який відповідає апіорному твердженню про те, що всі коефіцієнти регресії повинні бути невеликими числами, ідеально – більшість малозначних коефіцієнти повинні бути нулями. Підставивши щільність цього апіорного розподілу в формулу вище, і прологарифмувавши, отримаємо наступну оптимізаційну задачу:

$$\sum_{i=1}^m \log P\{y^{(i)} \mid x^{(i)}, \theta\} - \lambda \|\theta\|^2 \rightarrow \max \quad (1.25)$$

де $\lambda = \text{const} / \sigma^2$ — параметр регуляризації.

Цей метод відомий як L2-регуляризована логістична регресія, так як в цільову функцію входить L2-норма вектору параметрів для регуляризації.

Якщо замість L2-норми використовувати L1-норму, що еквівалентно використанню розподілу Лапласа, як апріорного, замість нормального, то вийде інший поширений варіант методу - L1-регуляризована логістична регресія:

$$\sum_{i=1}^m \log P\{y^{(i)} | x^{(i)}, \theta\} - \lambda \|\theta\|_1 \rightarrow \max \quad (1.26)$$

1.2.4 Дерева рішень

Дерево прийняття рішень (також може називатися деревом класифікації або регресійний деревом) - засіб підтримки прийняття рішень, який використовує деревовидний граф [7]. Даний метод використовується в статистиці і аналізі даних для створення прогнозуючих моделей. Структура дерева являє собою "листя" і "гілки". На ребрах («гілках») дерева рішення записані атрибути, від яких залежить цільова функція, в «листі» записані значення цільової функції, а в інших вузлах - атрибути, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листа і видати відповідне значення.

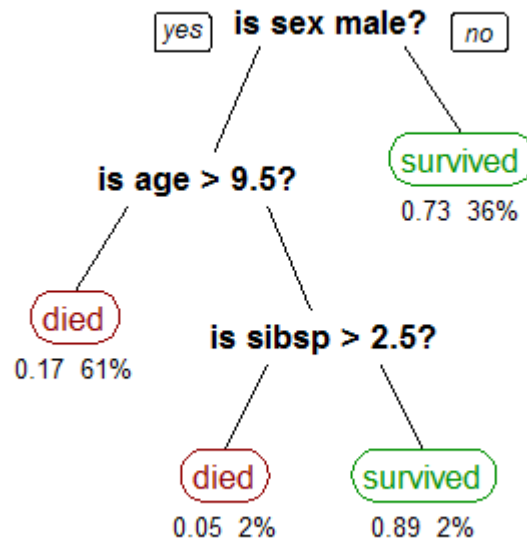


Рисунок 1.7 – Схематичне зображення дерева рішень

Подібні дерева рішень широко використовуються в інтелектуальному аналізі даних, а також в машинному навчанні. Мета полягає в створенні моделі, яка передбачає значення цільової змінної на основі декількох змінних на вході.

Кожен лист представляє собою значення цільової змінної, зміненої в ході руху від кореня по листу. Кожен внутрішній вузол відповідає одній з вхідних змінних. Дерево може бути також «вивчено» поділом вихідних наборів змінних на підмножини, засновані на тестуванні значень атрибутів. Це процес, який повторюється на кожній з отриманих підмножин. Рекурсія завершується тоді, коли підмножина в вузлі має ті ж значення цільової змінної, таким чином, воно не додає цінності для прогнозування. Процес, що йде «зверху вниз», є прикладом поглинає «жадібного» алгоритму, і на сьогоднішній день є найбільш поширеною стратегією дерев рішень для даних, але це не єдина можлива стратегія. В інтелектуальному аналізі даних, дерева рішень можуть бути використані в якості математичних і обчислювальних методів, щоб допомогти описати, класифікувати і узагальнити набір даних, який може бути записаний таким чином:

$$(x, Y) = (x_1, x_2, x_3 \dots x_k, Y) \quad (1.27)$$

Залежна змінна Y є цільовою змінною, яку необхідно проаналізувати, класифікувати і узагальнити. Вектор x складається з вхідних змінних x_1, x_2, x_3 і т. д., які використовуються для виконання даного завдання.

При аналізі рішень «дерево рішень» використовуються як візуальний і аналітичний інструмент підтримки прийняття рішень, де розраховуються очікувані значення (або очікувана корисність) конкуруючих альтернатив.

Дерево рішень складається з трьох типів вузлів:

- вузли рішення - зазвичай представлені квадратами;
- імовірнісні вузли - представляються у вигляді кола;
- замикаючі вузли - представляються у вигляді трикутника.

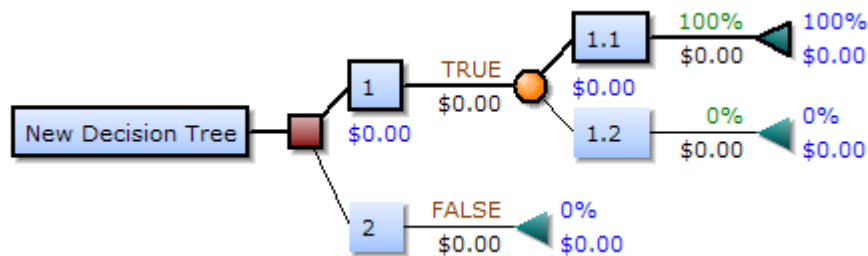


Рисунок 1.8 – Візуальне представлення дерева рішень [7]

На рисунку 1.8, дерево рішень слід читати зліва направо. Дерево рішень не може містити в собі циклічні елементи, тобто кожен новий лист згодом може лише розщеплюватися, відсутні сполучні шляхи. Зазвичай дерево рішень представляється у вигляді символічної схеми, завдяки якій його простіше сприймати і аналізувати.

Дерева рішень бувають двох основних типів:

- дерево для класифікації, коли що передбачається результат є класом, до якого належать дані;

- дерево для регресії, коли передбачається, що результат можна розглядати як дійсне число (наприклад, ціна на будинок, або тривалість перебування пацієнта в лікарні).

1.2.4 Нейронні мережі

Нейронні мережі – логічні структури, які представляють собою послідовність нейронів з'єднаних між собою синапсами [8]. Нейронні мережі, в залежності від реалізованої архітектури, дають можливість вирішувати задачі як класифікації, так і регресії.

Нейрон - це обчислювальна одиниця, яка отримує інформацію, виробляє над нею прості обчислення і передає її далі. Вони діляться на три основних типи: вхідний, прихований і вихідний, також є нейрон зміщення і контекстний нейрон. У випадку коли нейронна мережа складається з великої кількості нейронів, вводять термін шару. Відповідно, є вхідний шар, який отримує інформацію, n прихованих шарів, які її обробляють і вихідний шар, який виводить результат. У кожного з нейронів є 2 основні параметри: вхідні дані і вихідні дані. У разі вхідного нейрона вхідні дані дорівнюють вихідним. В інших, у якості вхідних даних потрапляє сумарна інформація всіх нейронів з попереднього шару, після чого, вона нормалізується, за допомогою функції активації і передається на наступний шар.

Синапс – це зв'язок між двома нейронами. У синапсів є 1 параметр - вага. Завдяки йому, вхідна інформація змінюється, коли передається від одного нейрона до іншого. В ході тренування ці ваги змінюються для мінімізації цільової функції, що являє собою процес навчання нейронної мережі.

На рисунку 1.9 зображена схема з'єднання синапсів та нейронів, на якій у вигляді кругів зображені нейрони, у вигляді прямих ліній – синапси з відповідними вагами.

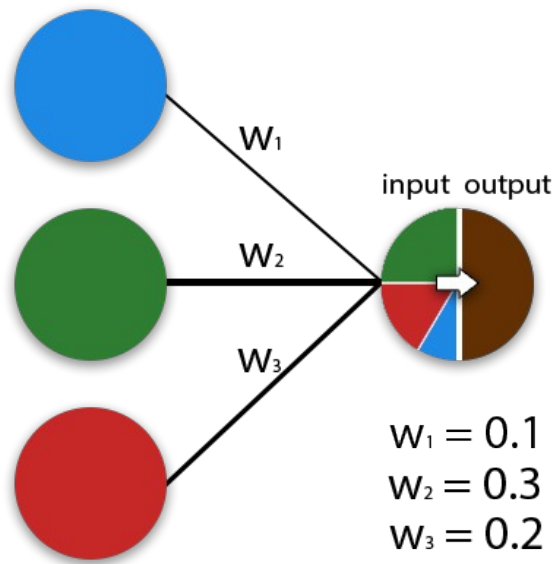


Рисунок 1.9 - Схема з'єднання синапсів та нейронів [8]

Функція активації - це спосіб нормалізації вхідних даних, завдяки якій на виході з нейрону отримується значення у потрібному числовому значенні. Функцій активації досить багато тому ми розглянемо найосновніші: лінійна, сигмоїда і гіперболічний тангенс. Головні їх відмінності - це діапазон значень.

1.3 Метрики оцінювання

Метрики оцінювання можна умовно поділити на два класи: методи оцінки якості прогнозування алгоритму та метрики оцінки якості ранжування.

Метрика точності прогнозування «accuracy» рахує відносну кількість правильних прогнозувань:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i) \quad (1.28)$$

де n – розмір вибірки;

y – цільове значення;

\hat{y} – прогнозоване значення.

Недоліком цієї метрики є те, що у разі оцінки якості класифікатора ми не можемо у точній мірі визначити його якість, оскільки дана метрика не враховує значення ймовірності.

Для більш точної оцінки використовується метрика середньоквадратичної помилки «RMSE» [9]:

$$\text{RMSE}(\hat{\theta}) = \sqrt{\text{MSE}(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)} \quad (1.29)$$

де $\hat{\theta}$ - прогнозоване значення;

θ - цільове значення.

Передумова метрики «DCG» полягає в тому, що найрелевантніші документи, які знаходяться нижче в списку результатів пошуку, повинні бути оштрафовані, оскільки оцінюване значення релевантності зменшується у логарифмічному порядку пропорційно положенню результату [10]. Метрика «DCG» визначається як:

$$\text{DCG}_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i+1)} \quad (1.30)$$

де rel_i – оцінка релевантності результату на позиції i .

Списки результатів пошуку можуть відрізнятися за довжиною залежно від запиту. Співставлення продуктивності пошукової системи від одного запиту до наступного не може бути послідовно досягнуто з використанням тільки метрики «DCG», тому кумулятивний коефіцієнт посилення в кожній позиції для обраного значення p повинен бути нормалізований по запитам, для цього додатково використовується метрика «nDCG»:

$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p} \quad (1.31)$$

де «IDCG» обраховується за наступною формулою:

$$\text{IDCG}_p = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \quad (1.32)$$

1.4 Функції активації

1.4.1 Функція сигмоїди

Функція сигмоїди - це математична функція, що має характерну S-подібну крив.

Сигмоїда визначена наступною формулою:

$$S(x) = \frac{1}{1 + e^{-x}} \quad (1.33)$$

Значення на виході з сигмоїди зображені на рисунку 1.10.

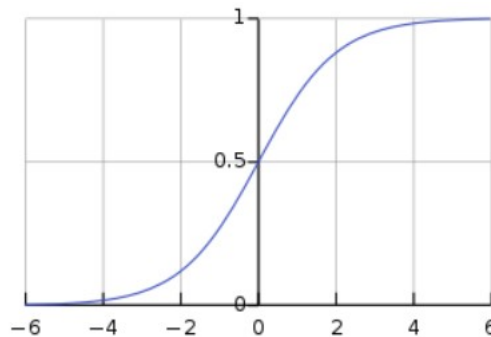


Рисунок 1.10 — Розподіл значень на виході функції сигмоїди

1.4.2 Тангенціальна функція

На відміну від сигмоїди, тангенціальна функція активації дозволяє на виході отримати значення в діапазоні від -1 до 1, що в деяких випадках спрощує процес навчання нейронної мережі.

Тангенціальна функція визначена як:

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}. \quad (1.33)$$

Розподіл значень тангенціальної функції зображений на рисунку 1.11.

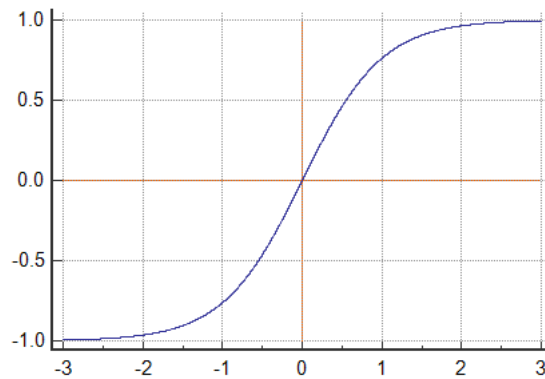


Рисунок 1.11 — Розподіл значень тангенціальної функції

1.4.3 Softmax

Функція Softmax, або ж нормована експоненційна функція - це узагальнення логістичної функції, що "стискує" K -вимірний вектор із довільним значеннями компонент до K -вимірного вектора $\sigma(\mathbf{z})$ з дійсними значеннями компонент в області $[0, 1]$ що в сумі дають одиницю. Функція задається наступним чином:

$$\sigma : \mathbb{R}^K \rightarrow [0, 1]^K$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K. \quad (1.34)$$

1.5 Вимоги до системи

Проаналізувавши наявні системи пошуку роботи та базові алгоритми підбору вакансій, необхідно сформулювати набір функціональних і нефункціональних вимог до системи.

1.5.1 Функціональні вимоги

До системи висуваються наступні функціональні вимоги:

- можливість пошуку вакансій;
- можливість перегляду вакансій;
- можливість налаштування різних фільтрів при пошуці роботи, як наприклад: місто, регіон, зарплата і т.д.;
- можливість підписки на розсилку вакансій по заданому пошуковому запиту на електронну пошту;
- можливість зареєстровану користувачу зберігати обрані вакансії;
- можливість відкликатися на вакансію.

1.5.2 Нефункціональні вимоги

До системи висуваються наступні нефункціональні вимоги:

- точність підбору вакансій не менше ніж 50%;
- вищу точність підбору вакансій ніж у базових алгоритмів;
- час підбору вакансій не більше 500 мілісекунд;
- модуль підбору вакансій повинен неперервно навчатися на основі зібраних історичних даних ;
- архітектура системи повинна враховувати можливість легкої інтеграції з різноманітними системами пошуку вакансій.

Підсумовуючи, в даному розділі були розглянуті систему пошуку вакансій, а також базові алгоритми машинного навчання, за допомогою яких можна вирішувати задачу підбору вакансій.

Сильні ринкові позиції розглянутих систем пошуку вакансій свідчать про актуальність нових рішень, які дають можливість підвищити якість сервісу, і як наслідок, підвищити конверсію.

Основна різниця між методами, які дозволяють вирішити задачу підбору вакансій полягає у наступному:

- складність реалізації;

- інтерпретація прогнозованих значень;
- швидкість прогнозування;
- змога знаходження закономірностей у вхідних даних.
- якість прогнозування.

Логістична регресія дозволяє вирішити задачу підбору вакансій. До переваг вказаного методу можна віднести наступне:

- легкість реалізації;
- проста інтерпретація роботи;
- швидкість роботи.

До недоліків логістичної регресії можна віднести:

- обмеженість в реалізації (не має можливості налаштувати алгоритм на вирішення конкретної задачі);
- обмеженість у використанні (в деяких випадках логістична регресія не здатна знайти закономірність у вхідних даних).

Дерева рішень також є одним із методів, які вирішуються дану задачу.

Алгоритм має наступні переваги:

- здатність до знаходження складних закономірностей в даних;
- велика кількість параметрів, які дозволяють налаштувати роботу алгоритму на вирішення конкретної задачі;
- велика кількість варіацій та різних реалізацій роботи цього методу (RandomForest, LightGBM, XgBoost, AdaBoost і т.д).

Недоліки дерев рішень:

- схильність до перенавчання, як наслідок складної архітектури;
- ускладнення архітектури сприяє збільшенню часу прогнозування.

Нейронні мережі мають наступні переваги:

- велика варіативність реалізацій (рекурентні, згорткові, рекурсивні мережі);
- здатні знаходити складні закономірності в даних;
- варіативність параметрів моделі, які дозволяють регуляризувати модель.

Недоліки нейронних мереж:

- складність інтерпретації роботи моделі;
- збільшення рівнів в нейронній мережі сприяє зростання часу прогнозування.

В даній роботі розглядаються вказані вище методи, а також проводиться порівняльний аналіз роботи моделей для знаходження найкращого варіанту реалізації моделі в системі підбору вакансій.

2 МАТЕМАТИЧНА МОДЕЛЬ

2.1 Постановка задачі

В даній роботі задача підбору вакансій постає у наступному вигляді:

1. На вхід подається список вакансій фіксованої довжини, кожна з яких трансформується в числовий вектор на основі свої характеристик.
2. На основі історичних даних, кожній вакансії у відповідність ставиться цільове значення 1 або 0, тобто чи була вакансій обрана в межах одного пошукового запиту, чи ні.
3. Далі, на основі цільового значення модель навчається прогнозувати вірогідність вибору вакансії в межах одного пошукового запиту.
4. На сумарну вірогідність вибору всіх вакансій в межах одного пошукового запиту накладається обмеження — вона повинна дорівнювати 1.
5. Відповідно, отримавши вірогідність вибору для кожної вакансій в межах одного пошукового запиту, кожна з вакансій сортується по вірогідності за спаданням.

Таким чином, математична модель підбору вакансій прогнозує значення для кожної з вакансій в межах одного пошукового запиту в діапазоні від 0 до 1, відповідно робиться наступне припущення: чим більше значення вірогідності — тим більше вакансія має підходити користувачу.

З огляду на це — для вирішення даної задачі в основу математичної моделі закладені нейронні мережі, оскільки даний алгоритм машинного навчання дозволяє охоплювати контекст не лише одного одного зразка вхідних даних, в даному випадку вхідний зразок — характеристики вакансій в числовому векторі. Тобто, з'являється можливість охоплювати контекст зразків вхідних даних довільної довжини, що й необхідно для вирішення поставленої задачі.

2.2 Skip-gram

В даній роботі використовуються текстові дані, які потрібно певним чином трансформувати в векторне числове представлення, для використання їх у якості вхідних факторів для моделі. Для вирішення цього питання використовується модель Skip-gram.

Модель Skip-gram намагається максимізувати класифікацію слова на основі іншого слова в реченні. Точніше, використовується кожне поточне слово як вхід для логарифмічного лінійного класифікатора з безперервним проекційним шаром і прогножуються слова в певному заданому діапазоні до і після поточного слова. В свою чергу, збільшення діапазону покращує якість отриманих векторів слів, але також збільшує обчислювальну складність. Оскільки більш віддалені слова, як правило, менш пов'язані з поточним словом, ніж ті, які близькі до нього, віддаленими словами віддається менше ваги [11].

Для покращення процесу навчання моделі, використовується генерація негативних прикладів, тобто беруться випадкові слова, які не зустрічались разом у вказаному діапазоні і маркуються моделі як 0, відповідно, слова, які зустрічались разом — маркуються як 1.

У якості цільової функції використовується наступна функція:

$$J_{\text{NEG}} = \log Q_{\theta}(D = \quad \quad \quad (2.1)$$

де $Q_{\theta}(D = 1|w, h)$ — бінарна логістична регресійна вірогідність моделі появи слова w в контексті h , в корпусі D .

Вказана функція максимізується, коли модель віддає високі вірогідності реальним словам, і малі вірогідності шумним словам.

На рисунку 2.1 зображена загальна архітектура моделі Skip-gram.

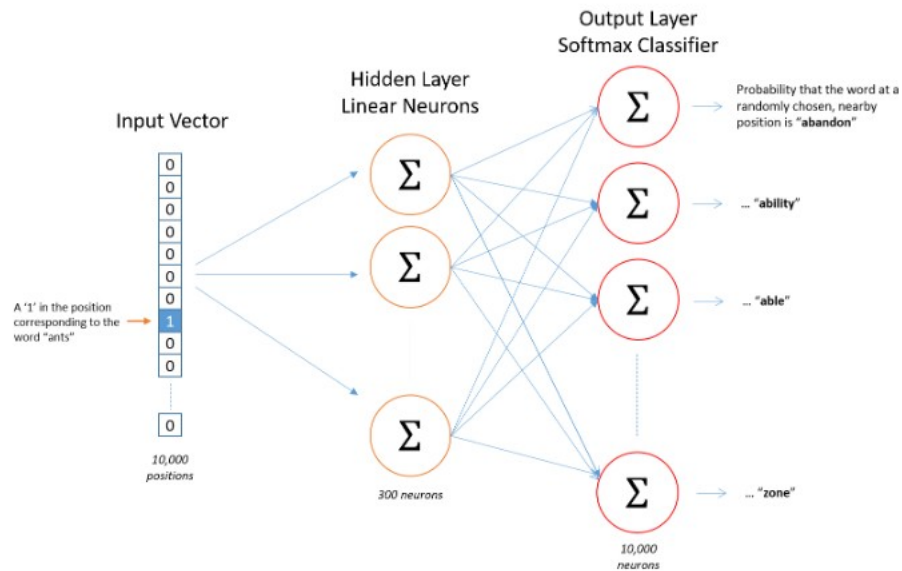


Рисунок 2.1 - Архітектура Skip-gram моделі [11]

Модель має наступні параметри:

- кількість епох (ітерацій) при навчанні;
- розмір діапазону контексту слова, який прогнозує дана модель;
- розмір прихованого шару, який використовується для переведення слова в векторний простір.

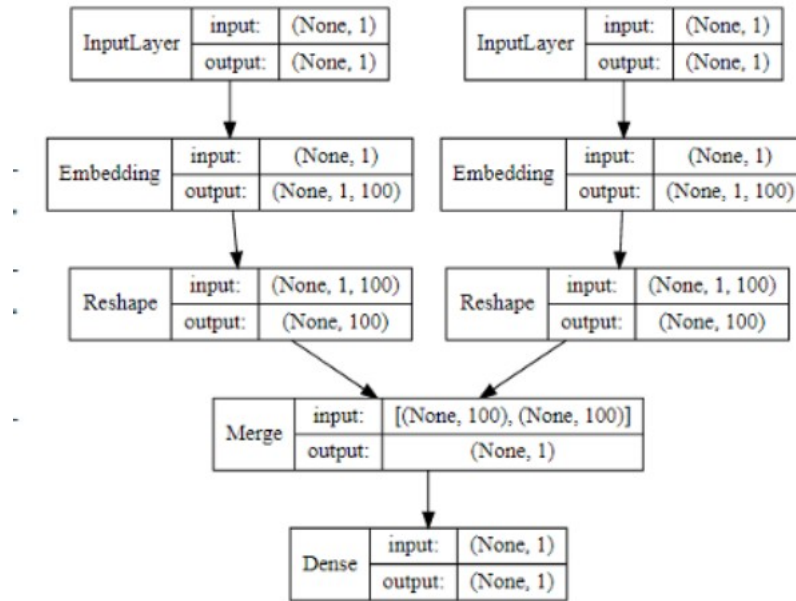


Рисунок 2.2 — Архітектура моделі побудована за допомогою фреймворку Keras

Як результат, для кожного слова може бути отриманий числовий вектор, який за певною функцією схожості (евклідова, косинусна відстань) буде близький до вектора слова схожого за значенням.

В базовій реалізації для переведення речення в числовий простір, для кожного слова отримується числовий вектор, після чого всі вектора слів в реченні додаються, та береться середнє — таким чином отримується числовий вектор речення. В даному випадку переведення речення в числовий простір позначається як s , та обчислюється за формулою:

$$s = \frac{1}{j} \sum_{i=1}^j e_i \quad (2.2)$$

де e_i — відповідний числовий вектор слова;

j — кількість слів у реченні.

На відміну від базової реалізації, в даній роботі для переведення речення в числовий вектор, окрім числового вектора слова, використовується інверсія частоти (IDF), з якою слово зустрічається в документах навчального корпусу. Це дозволяє зменшити вагу широкоживаних слів. Даний підхід дозволяє виділити характерні слова в реченні на фоні слів, які мають допоміжне значення. В даному випадку переведення речення в числовий простір позначається як s , та обчислюється за формулою:

$$s = (2.3)$$

де e_i – числовий вектор слова i ;

idf_i – інверсія частоти слова i .

Де idf_i обчислюється за наступною формулою:

$$idf_i = \log(N/df_i) \quad (2.4)$$

де N – загальна кількість документів в корпусі;

df_i – кількість документів, в яких зустрічається слово i .

На рисунку 2.3 наведений приклад результату роботи моделі Skip-gram, на виході слова схожі за значення розташовуються поряд у векторному числовому просторі.

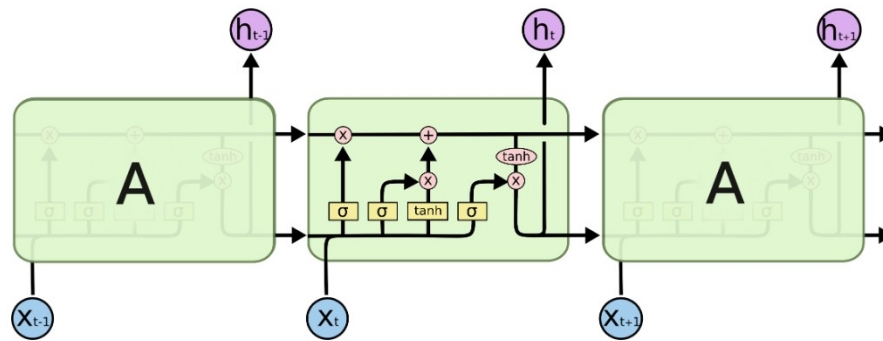


Рисунок 2.4 — Структура послідовно з'єднаних блоків LSTM [12]

Інформація від одного блоку до іншого передається у вигляді декількох змінних стану, як зображено на рисунку 2.5.

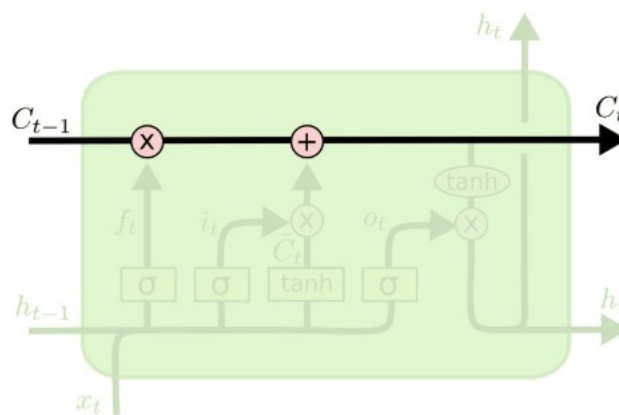


Рисунок 2.5 — Структура блоку LSTM [12]

c_t — змінна стану блоку, яка змінюється під час проходження через блок внаслідок лінійних взаємодій з іншими змінними через так звані затвори у вигляді шару нейронної мережі з сигмоїдною функцією активації, які додають або видаляють певну інформацію до змінної стану.

При подачі даних на вхід визначається яку частину інформації потрібно викинути зі змінної стану блоку. Це рішення приймає сигмоїдальний шар мережі

f_t , якому на вхід подається h_{t-1} та x_t . На виході отримується число в діапазоні від 0 до 1, що відповідно значить зберегти або видалити цю інформацію [12].

$$f_t = \sigma(W \quad (2.5)$$

де W_f , b_f – ваги шару f_t ;

h_{t-1} – прихований стан попереднього блоку;

x_t – вхідні данні.

Наступний крок — визначення того, яку інформацію зберегти в змінні стану. Даний крок складається з двох частин: спочатку сигмоїдальний шар i_t вирішує які значення оновлювати далі, тангенціальний шар створює вектор нових потенційних значень \tilde{C}_t , які можуть бути додані до змінної стану.

$$\begin{aligned} i_t &= \sigma(W_i \cdot \\ \tilde{C}_t &= \tanh(V \quad (2.6) \end{aligned}$$

Далі оновлюється значення змінної стану на основі раніше обрахованих змінних, які дозволяють видалити стару або ж додати нову інформацію:

$$C_t = f_t \quad (2.7)$$

На кінцевому етап обраховується значення, яке буде подаватися на вихід. Це значення базується на змінній стану. Спочатку, на вхід сигмоїдальному шару o_t подається значення змінної стану c_t , це дозволяє визначити яку інформацію слід залишити, а яку відфільтрувати. Далі, значення змінної стану подається на вхід тангенціального шару h_t і перемножується з раніше отриманим значення o_t .

$$\begin{aligned} o_t &= \sigma(V \\ h_t &= o_t * \quad (2.8) \end{aligned}$$

Отримані значення h_t для кожного блоку надходять далі для визначення того, якому блоку слід виділити більше уваги.

Параметри моделі:

В роботі на вхід LSTM подається вектор розмірністю $s \times m \times n$

де s – об'єм навчальної вибірки $(\sim 3m)$;

m – кількість факторів;

n – кількість результатів в межах одного пошуку $(=20)$.

У якості вихідних даних, які прогнозує модель, кожному результату в пошуку у відповідність ставиться 0 або 1 (користувач обрав дану вакансію, чи ні)

Розмірність вектору прихованого стану: 100.

Функція активації останнього шару — сигмоїда.

Функція втрат при навчанні моделі: бінарна перехресна ентропія, яка обчислюється за формулою:

$$H(p, q) = (2.9)$$

де $p(x)$ – істинне значення;

$q(x)$ – прогнозоване значення.

В даній роботі до базової реалізації моделі LSTM був доданий додатковий шар Attention Mechanism, який дозволяє отримати значення вагів уваги до кожного вектору вхідних даних. Іншими словами, за допомогою такої модифікації ми можемо обчислювати значення вагів уваги для кожної вакансії в межах пошукового запиту, та використовувати це значення для ранжування пошукових результатів в межах пошуку. Принцип роботи, та реалізація шару Attention Mechanism приведена в розділі 2.4.

2.4 Attention Mechanism

Механізми уваги забезпечують більш пряму залежність між станом моделі в різні моменти часу, таким чином можна прослідкувати якій вхідній послідовності було виділено більше уваги. Це дає змогу отримати результат роботи моделі в іншому векторному просторі, що може бути використано для подальшого аналізу результатів. Для визначення цієї залежності розраховується контекстний вектор c_t , на основі значення прихованого значення стану моделі h_t [13].

$$c = \sum_{t=1}^T \alpha_t h_t \quad (2.10)$$

де T — кількість кроків в часі у вхідній послідовності;

α_t — ваги розраховані для в кожен момент часу t для прихованого значення стану моделі h_t .

Ваги розраховуються за наступною формулою:

$$e_t = a(h_t),$$
$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^T \exp(e_k)} \quad (2.11)$$

де a — навчальна функція, яку можна розглядати як обчислення скалярного значення важливості для прихованого значення h_t .

На рисунку 2.6 зображена структура механізму уваги Attention Mechanism.

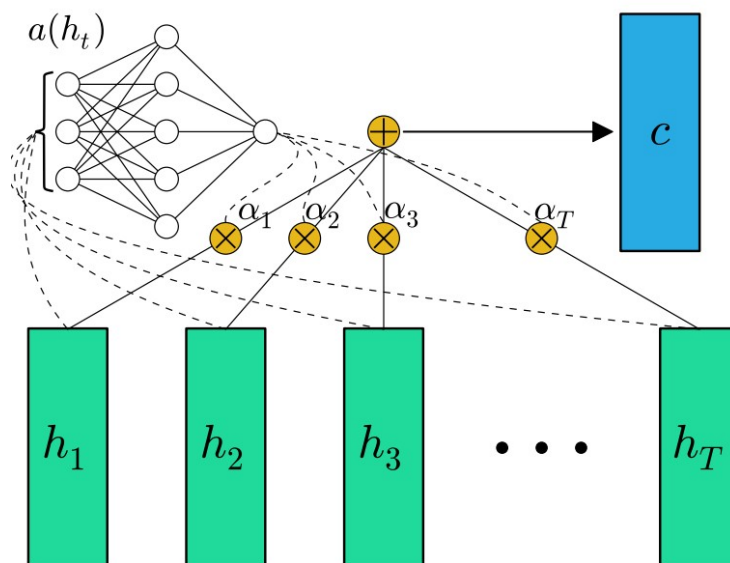


Рисунок 2.6 — Структура механізму уваги Attention Mechanism [13]

На виході з моделі ми отримуємо скалярні значення уваги для кожного пошукового результату, відповідно, сумарні значення уваги для одного пошуку дорівнюють одиниці.

Структурна схема нейронної мережі LSTM + Attention Mechanism наведена в додатку А.

На рисунку 2.7 зображено приклад розподілення уваги між вакансіями в межах одного пошукового запиту. На осі абсцис відкладено порядковий номер вакансій починаючи з 0, на осі ординат відповідне значення уваги для кожної вакансії.

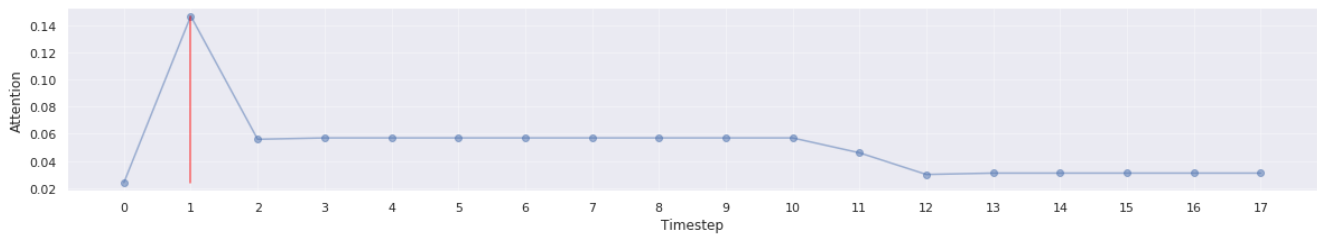


Рисунок 2.7 — Приклад розподілення уваги між вакансіям в межах одного пошукового запиту

2.5 Latent Semantic Model

Модель Latent Semantic Model призначена для переведення пошукових запитів та інформацію про вакансію в один семантичний простір, що дозволяє згенерувати нові фактори в якості вхідних даних для моделі ранжування.

На вхід моделі подаються наступні дані: пошуковий запит, заголовок вакансії, яка була переглянута у межах пошукового запиту, та n заголовків вакансій, які не були переглянуті в межах того самого пошукового запиту. Такий формат вхідних даних дозволяє вилучити з пари пошуковий запит — заголовок характерну семантичну інформацію.

На відміну від базової реалізації Convolutional Latent Semantic Model [14], в моделі, яка була розроблена в даній роботі, послідовність шару перетворення тексту в n -грамми, та згорткового шару були замінені на шар переведення тексту в векторний числовий простір за допомогою моделі Skip-gram. Ця модифікація дозволила подати на вхід моделі не лише значення тексту, але й можливий контекст, який враховувався при навчанні моделі Skip-gram, крім того, це суттєво зменшило час навчання нейронної мережі.

Вхідні текстові дані переводяться у векторний числовий простір, за допомогою моделі Skip-gram, на виході отримуємо вектор v .

На наступному кроці використовується шар нелінійної трансформації з тангенціальною функцією активації для вилучення семантичного представлення, визначеного як y .

$$y = \tanh(W_s \cdot v) \quad (2.12)$$

де v — вхідний вектор тексту пошукового запиту, або документу отриманого на виході з Skip-gram моделі;

W_s — матриця семантичної проекції;

y — представлення вхідного запиту або документа в латентному семантичному просторі з розмірністю L .

З урахуванням пошукового запиту та набору документів, які потрібно класифікувати, спершу обчислюється семантичне векторне представлення y для запиту та всі документи, що використовують LSM, як описано вище. Далі, обчислюється оцінка відповідності між запитом та кожним документом, вимірюючи косинусну схожість між їхніми семантичними векторами. Формально, семантична оцінка відповідності між запитом і документом визначається як:

$$R(Q, D) = \text{cosine}(y_Q, y_D) = \frac{y_Q^T y_D}{\|y_Q\| \|y_D\|} \quad (2.13)$$

де y_q та y_d — семантичні вектори пошукового запиту та документу, відповідно.

Навчальний набір даних, який використовувався в даній роботі складається з пошукових запитів, та документів, які були переглянуті під час пошуку, а також випадково вибраних документів, які не були переглянуті, в межах тих самих пошукових запитів. Відповідно, спочатку, ми отримуємо оцінку відповідності документа до заданого пошукового запиту, та перетворюємо дану оцінку в апостеріорну ймовірність даного документа за допомогою softmax функції:

$$P(D^+|Q) = \frac{\exp(R(Q, D^+))}{\sum_{D' \in D} \exp(R(Q, D'))} \quad (2.14)$$

де D – набір документів, для оцінки схожості до пошукового запиту, який складається з переглянутого документу $\{D_j^-; j = 1, \dots, J\}$. D_+ та набору з J не переглянутих документів;

(Q, D^+) – пара пошукового запиту та переглянутого документу.

В якості функції витрат використовується функція максимізації правдоподібності переглянутого документу з заданим пошуковим запитом в навчальній вибірці. Оскільки модель навчається у процесі мінімізації функції витрат, тому функція витрат приймає наступний вигляд:

$$L(\Lambda) = -\log \prod_{(Q, D^+)} P(D^+ | Q) \quad (2.15)$$

де Λ — визначається як набір мета-параметрів моделі.

Отже, в даному розділі були визначені та розглянуті математичні моделі, які будуть застосовуватися в системі підбору вакансій.

Було сформовано постановку задачі, які буде вирішуватися математичними моделями розглянутими в даному розділі.

Для представлення текстів у числовому просторі застосовується модель Skip-gram з модифікованим алгоритмом перетворення речення у числовий простір, який враховує не лише контекст кожного слова, а також його частотні характеристики.

В основі математичної моделі лежить реалізація LSTM з механізмом уваги (Attention Mechanism), яка дозволить визначити скільки уваги варто призначити кожній вакансії в межах пошукового запиту, для кінцевого підбору вакансій.

Додатково для представлення тексту пошукового запиту та заголовку вакансій була розроблена модифікована версія моделі Latent Semantic Model, яка дозволить представити пошуковий запит та вакансій у семантичному просторі, що

дає можливість оцінити за косинусною мірою релевантність вакансії до пошукового запиту.

Структурна схема нейронної мережі Latent Semantic Model наведена в додатку Б.

В розділі 4 буде розглянуто ефективність розроблених методів у порівнянні з базовими методами машинного навчання, що дасть точну оцінку ефективності розроблених математичних моделей.

3 ТЕХНОЛОГІЧНІ ТА АРХІТЕКТУРНІ АСПЕКТИ СИСТЕМИ

3.1 Архітектура системи

3.1.1 Мікросервісна архітектура

Мікросервіси — це технологія розробки програмного забезпечення, варіант архітектурного стилю сервіс-орієнтованої архітектури (SOA), який структурує додаток як сукупність сервісів, які слабо пов'язані між собою. В архітектурі мікросервісів послуги є певним набором простих функцій, з використанням економічних мережевих протоколів, як наприклад, REST. Перевага розкладання додатки на кілька менших сервісів полягає в тому, що він покращує модульність.

Це спрощує розуміння, розробку, тестування додатку, а також сприяє підвищенню стійкості до ерозії архітектури. Даний підхід дозволяє запровадити процес паралельної розробки, дозволяючи невеликим автономним командам самостійно розробляти, розгортати і масштабувати свої відповідні послуги.

Архітектури на основі мікросервісів забезпечують безперервну доставку і розгортання.

Мікросервіси мають наступні властивості:

- Послуги в архітектурі мікросервіса (MSA) часто є процесами, які здійснюють зв'язок по мережі для досягнення мети з використанням мережевих протоколів, таких як HTTP. Однак служби можуть також використовувати інші види механізмів взаємодії між процесами, таких як колективна пам'ять;
- Послуги в архітектурі мікросервіса незалежно розгортаються;
- Послуги легко замінити;
- Послуги організовані навколо можливостей, наприклад, інтерфейсу користувача, рекомендацій, логістики, виставлення рахунків і т. д.;

- Послуги можуть бути реалізовані з використанням різних мов програмування, баз даних, апаратного і програмного середовища, в залежності від того, що підходить найкраще;
- Послуги невеликі за розміром, підтримка обміну повідомленнями, обмежена контекстами, автономно розроблена, незалежно розгортається, децентралізована та побудована і випущена з автоматизованими процесами.

Архітектура на базі мікросервісів має наступні властивості:

- Природно забезпечує модульну структуру;
- Забезпечує безперервний процес розробки програмного забезпечення для розгортання. Перехід на невелику частину додатка вимагає тільки відновлення і повторного розгортання тільки одного або невеликої кількості сервісів;
- Дотримується таких принципів, як дрібнозернисті інтерфейси, бізнес-орієнтована розробка, архітектури хмарних додатків, програмування.

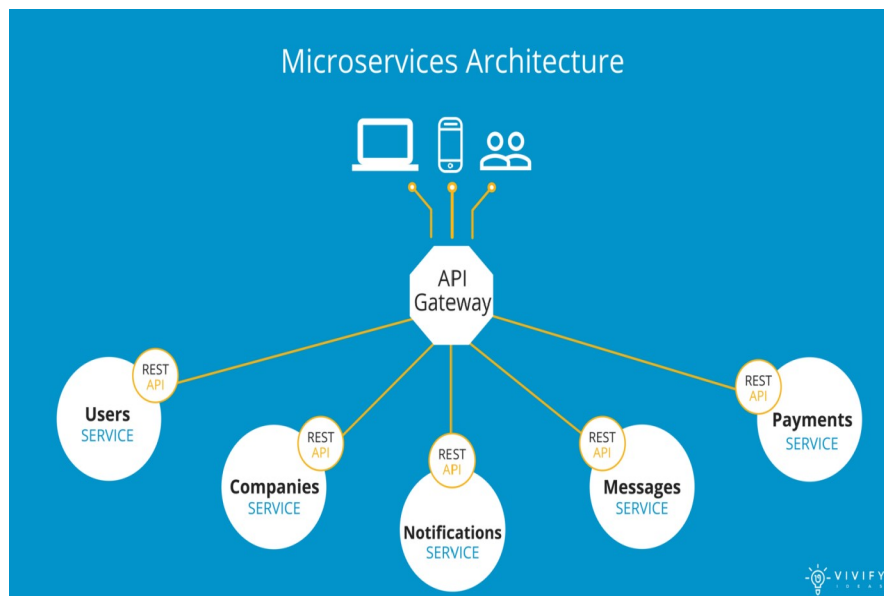


Рисунок 3.1 – Приклад реалізації мікросервісної архітектури [15]

3.1.2 Сервіси системи

Внаслідок мікросервісної декомпозиції, система підбору вакансій на основі історичних даних має наступні сервіси:

- сервіс пошуку вакансій
- сервіс розсилки
- сервіс збору статистичних даних
- сервіс індексації вакансій
- сервіс обробки вакансій
- сервіс підбору вакансій
- сервіс навчання моделей
- сервіс відклику на вакансію

3.1.2.1 Сервіс пошуку вакансій

Даний сервіс відповідає за обробку пошукових запитів користувачів, та взаємодіє базою даних вакансій, а також з сервісом підбору вакансій для ранжування результатів на основі історичних даних.

Діаграма послідовності пошуку вакансій наведена в додатку В.

Сервіс відповідає за наступні функції:

- формування запиту в базу даних вакансій на основі заданого пошукового запиту;
- витягування відповідних вакансій у відповідь на пошуковий запит, або ж розсилку;
- відсилення отриманих результатів на сервіс підбору вакансій;
- формування кінцевого списку вакансій у якості відповіді користувачу;
- збереження у базу даних історії пошукових запитів.

3.1.2.2 Сервіс розсилки

Даний сервіс відповідає за управління розсилкою кінцевим користувачам. В свою чергу, сервіс взаємодіє з сервісом пошуку вакансій.

Сервіс відповідає за наступні функції:

- підписка користувачів на відповідний пошуковий запит;
- відписка користувачів на відповідний пошуковий запит;
- збереження у базу даних історії підписки користувачів на пошукові запити;
- відправлення користувачам на електронну пошту списку вакансій на заданий пошуковий запит.

3.1.2.3 Сервіс збору статистичних даних

Сервіс відповідає за збір статистичних даних, та їх збереження у базу даних.

Сервіс відповідає за наступні функції:

- збір та збереження даних про історію перегляду та вибору вакансій в межах пошукових запитів;
- збір та збереження даних про історію переходів та вибору вакансій з розсилки;
- збір та збереження даних про історію відкликів користувачів на вакансії.

3.1.2.4 Сервіс індексації вакансій

Сервіс відповідає за створення та оновлення вакансій.

Діаграма послідовності пошуку вакансій наведена в додатку Г.

Сервіс відповідає за наступні функції:

- створення вакансії, та збереження її в базу актуальних вакансій;
- оновлення вакансії, та збереження відповідної інформації в базу даних;
- надсилання відповідної інформації сервісу обробки вакансій.

3.1.2.5 Сервіс обробки вакансій

Сервіс обробки вакансій відповідає за вилучення інформації з вакансії, та представлення її у вигляді числових характеристик. Даний сервіс взаємодіє з сервісом індексації вакансій.

Сервіс відповідає за наступні функції:

- обробку та представлення вакансії у вигляді числових характеристик, та запис відповідних даних в базу даних;
- обробку текстової інформації вакансій, як наприклад: текст та заголовок вакансії, та збереження відповідної інформації в базу даних.

3.1.2.6 Сервіс підбору вакансій

Сервіс підбору вакансій відповідає за ранжування пошукових результатів отриманих від сервісу пошуку вакансій, за допомогою моделей, які були навчені відповідним сервісом.

Сервіс відповідає за наступні функції:

- ранжування пошукових результатів;
- збереження та використання математичних моделей.

3.1.2.7 Сервіс навчання моделей

Сервіс навчання моделей відповідає за навчання математичних моделей на основі історичних даних.

Діаграма послідовності навчання моделей наведена в додатку Д.

Сервіс відповідає за наступні функції:

- витягування статистичних даних та інформацію про відповідні вакансії, необхідну для навчання моделей;
- збереження моделей;
- аналіз якості роботи моделей підбору вакансій.

3.1.2.8 Сервіс відклику на вакансію

Сервіс відповідає за процес відклику користувача на вакансію.

Сервіс відповідає за наступні функції:

- надсилання інформацію про відклик на вакансію користувачу на електронну пошту;
- надсилання роботодавцю сповіщення про відклик користувача на його вакансію;
- збереження інформації про відклик у базу даних.

3.2 Структурна схема системи

Загалом, система налічує 8 сервісів, які безперервно взаємодіють один з одним.

Сервіси взаємодіють між собою на основі протоколу HTTP, та за допомогою технології REST, описаній в розділі 3.3.

При пошуковому запиті сервіс пошуку вакансій формує запит в базу даних для формування кінцевого списку вакансій, які релевантні до отриманого пошукового запиту. Після отримання результатів з бази даних, оброблені результати відправляються на модуль підбору вакансій, який ранжує вхідні результати на основі розроблених математичних моделей навчених на історичних даних, які були описані в розділі 2. Результат ранжування відправляється у якості відповіді на запит сервісу пошуку вакансій, та в кінцевому результаті відображається кінцевому користувачу.

Аналогічним чином працює сервіс розсилки, тобто, з заданою періодичністю, сервіс відправляє запит на сервіс пошуку вакансій з відповідним пошуковим запитом, після запит надсилається сервісу підбору вакансій, та по зворотньому ланцюгу повертається сервісу розсилки, після чого розсилка з відповідними пошуковими результатами відправляється кінцевому користувачу.

Результати взаємодії користувача з пошуковою видачею, відклику на вакансію та результати взаємодії з розсилкою детально зберігаються в базу даних

призначену для зберігання статистичних даних, які потім використовуються для навчання математичних моделей для підбору вакансій на основі історичних даних.

В момент створення створення або оновлення інформації про вакансію сервіс індексації вакансій відправляє відповідні дані про вакансію на сервіс обробки вакансій, який трансформує характеристики вакансії в числовий вектор, а також трансформує вектор тексту вакансій в числовий вектор, після чого відповідні дані зберігаються в базу даних для подальшого використання сервісом підбору вакансій.

Сервіс навчання моделей взаємодіє відразу з декількома сервісами. В заданій періодичності сервіс зчитує історичні дані за допомогою сервісу збору статистичних даних, трансформує отримані дані в формат необхідний для навчання моделей, та “донавчає” математичні моделі, які використовуються модулем підбору вакансій.

Сервіс підбору вакансій при отриманні запиту від сервісу пошуку вакансій ініціалізує математичну модель, після чого трансформує характеристики отриманих вакансій в числовий вектор факторів для подачі на вхід моделі. З виходу моделі кожній вакансії ставиться у відповідність певне число, певна оцінка, за якою вони будуть сортуватися. Після сортування відповідні вакансії відправляються у відповідь сервісу пошуку вакансій з вказанням оцінки для кожної вакансії.

Структурна схема системи пошуку вакансій на основі історичних даних наведена в додатку Е.

3.3 Вибір та обґрунтування елементів та технологій

Для розробки сервісу навчання моделей було обрано мову програмування Python. Даний вибір базується на тому, що в даний час найбільша кількість бібліотек для роботи з даними, для побудови нейронних мереж були створені саме

на цій мові програмування. Це значно спрощує процес розробки та тестування рішень у сфері машинного навчання.

Python – це мова програмування з декількома парадигмами. Повністю підтримуються об'єктно-орієнтоване програмування та структуроване програмування, а багато його функції підтримують функціональне програмування і аспектно-орієнтоване програмування (в тому числі метапрограмування і метаоб'єкти). Багато інших парадигми підтримуються за допомогою розширень, включаючи дизайн за контрактом та логічне програмування.

Python використовує динамічну типізацію і комбінацію підрахунку посилань і збирача сміття, який працює у режимі циклів, для управління пам'яттю. Він також має динамічний дозвіл імен, який пов'язує імена методів і змінних під час виконання програми.

Для обробки даних, та представлення її для візуалізації використовувалася бібліотека Pandas. Дана бібліотека дозволяє легко працювати з великими об'ємами даних, що значно спрощує процес підготовки вхідних даних для математичних моделей.

Pandas – це програмна бібліотека, написана для мови програмування Python для обробки і аналізу даних. Зокрема, він пропонує структури даних і операції для обробки числових таблиць і часових рядів. Це безкоштовне програмне забезпечення, випущене під ліцензією BSD.

Бібліотека забезпечує наступні функції:

- інструменти для читання і запису даних між структурами даних в пам'яті і різними форматами файлів;
- узгодження даних і комплексна обробка відсутніх даних;
- перестановка і поворот наборів даних;
- накладення на основі міток, зручна індексація і підножина великих наборів даних;
- вставка і видалення стовпців структури даних;

- об'єднання і агрегація даних;
- забезпечує фільтрацію даних.

Для оцінки якості моделей, та побудови базових лінійних моделей, дерев рішень використовується бібліотека Scikit-learn. Вибір даної бібліотеки обумовлюється тим, що дана бібліотека працює у взаємодії з іншими бібліотеками для обробки даних, як наприклад: NumPy та Pandas.

Scikit-learn — це бібліотека, яка дозволяє використовувати алгоритми машинного навчання для мови програмування Python. Він включає в себе різні алгоритми класифікації, регресії і кластеризації, включаючи підтримку SVM, дерев рішень, логістичної та лінійної регресії, k-means і DBSCAN і призначений для взаємодії з чисельними і науковими бібліотеками, які використовуються в мові програмування Python, як наприклад, NumPy і SciPy.

Для обробки текстових даних була обрана бібліотека Gensim. Вказана бібліотека має широкий набір функцій для перетворення текстових даних в числове представлення, що дозволяє швидко трансформувати дані для подання на вхід математичним моделям.

Gensim — це надійний інструментарій моделювання векторних просторів з відкритим кодом, реалізований у Python. Він використовує NumPy, SciPy. Gensim спеціально призначений для обробки великих текстових колекцій, що використовують потокове передавання даних.

Для побудови архітектури нейронних мереж використовувалися дві бібліотеки: Tensorflow та Keras. Вибір цих бібліотек обумовлюється тим, що Keras надає простий та зрозумілий інтерфейс для роботи з Tensorflow. В свою чергу, бібліотека Tensorflow дозволяє будувати велику кількість варіацій архітектур нейронних мереж, а також має підтримку GPU, що значно пришвидшує процес навчання нейронних мереж.

TensorFlow — це бібліотека програмного забезпечення з відкритим вихідним кодом для високопродуктивних чисельних обчислень. Його гнучка архітектура

дозволяє легко розгортати обчислення на різних платформах (процесорах, графічних процесорах, TPU) і від настільних комп'ютерів до кластерів серверів до мобільних і периферійних пристроїв. Спочатку розроблений дослідниками і інженерами з команди Google Brain в організації AI Google, він має сильну підтримку для машинного навчання і глибокого навчання, а гнучке обчислювальне ядро обчислень використовується в багатьох інших наукових областях.

Keras – це високорівнева API нейронних мереж, написаний на Python і здатний працювати поверх TensorFlow, CNTK або Theano. Він був розроблений з упором на можливість швидкого експериментування.

Keras має наступні переваги:

- дозволяє легко і швидко створювати прототипи (завдяки зручності, модульності і розширюваності);
- підтримує як згорткові мережі, так і рекурентні мережі, а також комбінації цих двох;
- легко працює на процесорі і графічному процесорі.

Для збереження статистичних даних використовується база даних ClickHouse.

ClickHouse – це стовпчаста аналітична СУБД з відкритим кодом, що дозволяє виконувати аналітичні запити в режимі реального часу на структурованих великих даних.

ClickHouse використовує власний діалект SQL близький до стандартного, але містить різні розширення: масиви та вкладені структури даних, функції вищого порядку, імовірнісні структури, функції для роботи з URI, можливості для роботи з зовнішніми сховищами, спеціалізовані агрегатні функції, функціонал для приблизних обчислень, можливість створення збережених уявлень з агрегацією.

Для взаємодії сервісу навчання моделей з іншими сервісами був обраний фреймворк aiohttp. Це простий та легкий у розгортанні асинхронний HTTP

фреймворк для реалізації архітектури клієнт/сервер для мови програмування Python.

Основні переваги даного фреймворку:

- підтримує як клієнтську, так і серверну сторону протоколу HTTP;
- підтримує як клієнтські, так і серверні веб-сокети;
- надає реалізацію веб-серверу з проміжними мережами і маршрутизацією.

Для взаємодії між сервісами в межах мікросервісної архітектури використовується технологія REST.

REST розшифровується як Representational State Transfer (передача репрезентативного стану). REST є веб-стандартом архітектури та використовує HTTP-протокол для передачі даних. Ця архітектура обертається навколо ресурсів, де кожен компонент є ресурсом і ресурс отримує доступ за допомогою загального інтерфейсу з використанням стандартних методів HTTP.

RESTful сервіс працює за допомогою Uniform Resource Locator (URL). Це логічне ім'я, яке відокремлює ідентичність ресурсу від того що, приймається або відправляється.

Сервіси для обміну інформацією між собою використовують чотири основних HTTP методів, які використовуються при розробці RESTful системи:

1) GET

Найбільш поширений метод. Запит GET передає уявлення іменованих ресурсів від сервера до клієнта. Клієнту не обов'язково знати що-небудь про ресурс запитувача. Те, що він отримує назад це потік байтів з метаданими, які вказують як інтерпретувати отриману інформацію. В інтернеті, як правило, це «text/html» або «application/xhtml+xml». Одним з ключових моментів, запиту GET є те, що у результаті цього запиту, на стороні серверу не повинно бути будь-яких змін. Це принципово безпечний запит;

2) POST

POST використовується, коли клієнт не може передбачити ідентичність ресурсу. При створенні нових об'єктів не можна передбачити, як сервер буде називати ці ресурси. Саме тому уявлення ресурсу розміщується в певний обробник (наприклад, сервлет). Сервер приймає вхідні дані, перевіряє їх, перевіряє облікові дані користувача і т.д. Після успішної обробки, сервер повертає код відповіді на 201 HTTP з «Location» заголовком, який вказує місце розташування новоствореного ресурсу. У відповіді POST метод повинен підтвердити запит;

3) PUT

Метод PUT використовується, коли є можливість оновити ресурс повністю через певний ресурс. Наприклад, якщо відомо, що стаття знаходиться в <http://example.org/article/1234>, можливо ввести нове значення ресурсів цієї статті безпосередньо через PUT за цією адресою.

Якщо фактичне місцезнаходження ресурсів невідомо, наприклад, коли додається нова стаття, але не відомо де даний ресурс буде розміщений, можна використати POST метод, і таким чином дозволити серверу визначити фактичний URL;

4) DELETE

DELETE метод досить легко зрозуміти. Він використовується для видалення певного ресурсу, визначеного за допомогою URI.

У разі успішного видалення, повертається статус 200 (OK) HTTP разом з тілом відповіді, також можлива відповідь із поданням видаленого елемента (часто вимагає занадто багато трафіку). Або повертається статус HTTP 204 (без вмісту) без тіла відповіді.

Діаграма розгортання системи пошуку вакансій на основі історичних даних наведена в додатку Ж.

3.4 Сценарії використання системи

В системі передбачені дві ключові ролі: клієнт та роботодавець.

В системі для клієнтів передбачені наступні сценарії користування:

- реєстрація користувача;
- редагування особистої інформації;
- пошук вакансій з фільтрами (регіон, місто, зарплата і т.д.);
- підписка на розсилку по певному пошуковому запитувати;
- відклик на вакансію;
- збереження обраних вакансій;
- перегляд вакансій по компаніям.

В системі для роботодавців передбачені наступні сценарії:

- реєстрація роботодавця;
- створення та розміщення вакансій;
- оновлення інформації про вакансії;
- видалення вакансії;
- перегляд статистики переглядів та відкликів на вакансії.

Діаграма сценаріїв використання системи пошуку вакансій на основі історичних даних наведена в додатку И.

3.5 Інтерфейс користувача

Інтерфейс користувача відображає стан, та певні кількісні та часові характеристики кожного з сервісів.

На інтерфейсі зображена інформація про наступні сервіси:

- сервіс пошуку вакансій;
- сервіс підбору вакансій;

- сервіс індексації вакансій;
- сервіс обробки вакансій;
- сервіс розсилки;
- сервіс відклику;
- сервіс збору статистичних даних.

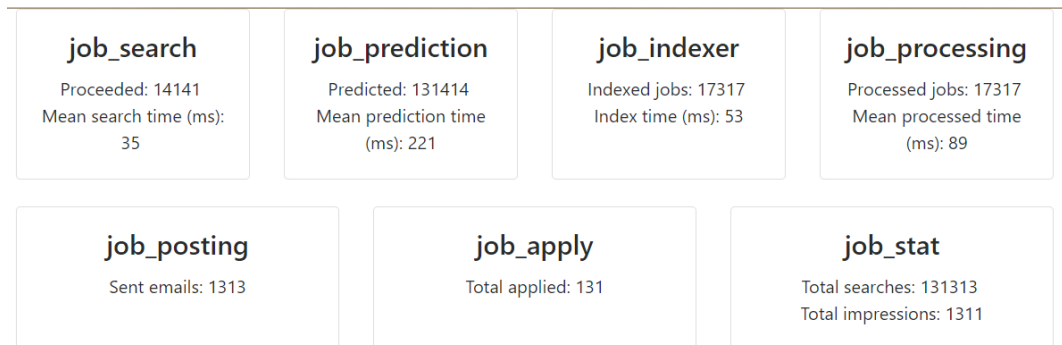


Рисунок 3.2 — Інтерфейс користувача системі підбору вакансій

3.6 Тестування системи

Для тестування були обрані наступні тест кейси:

- #1.Перевірка коректності підбору вакансій в межах пошукової видачі
- #2.Перевірка коректної конвертації характеристик вакансій в вектор числових значень
- #3.Перевірка коректної трансформації тексту вакансії в числовий вектор
- #4.Перевірка коректної десереалізація запиту отриманого від сервісу пошуку вакансій на сервісі підбору вакансій
- #5. Перевірка часу підбору вакансій на сервісі підбору вакансій

Відповідні тести кейси повинні задовільняти наступним умовам:

- #1. На виході з сервісу підбору вакансій кількість вакансій повинна відповідати вхідній кількості вакансій, які були надіслані з сервісу пошуку вакансій.
- #2. Характеристики вакансії повинні бути трансформовані у вектор заданої розмірності
- #3. Текст повинен бути трансформований у числовий вектор заданої розмірності
- #4. Задана структура отриманого запиту від сервісу пошуку вакансій повинна бути коректно десереалізована
- #5. Час підбору вакансій не повинен перевищувати 500 мс

Таблиця 3.1 — Матриця відповідності вимог

	Номер тесту	#1	#2	#3	#4	#5
Номер вимоги						
#1		+	+	+		
#2			+			
#3				+		
#4					+	
#5						+

Отже, в даному розділі було сформовано принцип побудови архітектури системи, а також була побудована структурна схеми системи підбору вакансій. Для побудови системи була обрана мікросервісна архітектура, що дозволить у майбутньому легко масштабувати систему, а також з'являється можливість легкої інтеграції системи у будь-які інші системи пошуку вакансій. Крім того, був

проведений вибір та обґрунтування елементів та технологій системи пошуку вакансій.

Загалом, система налічує 8 сервісів, які перебувають в безперервній взаємодії між собою. Також, система включає в себе дві основні бази даних, одна для збереження інформації для вакансій, інша — для збереження статистичних даних.

Було проведено успішне тестування системи, а саме, був визначений список відповідних тест-кейсів, у відповідність яким був поставлений список кінцевих умов.

4 ДОСЛІДЖЕННЯ АРХІТЕКТУРИ МАТЕМАТИЧНОЇ МОДЕЛІ

4.1 Методи порівняння алгоритмів

Для порівняння якості роботи алгоритмів можуть використовуватися наступні підходи:

- крос-валідація;
- оцінка якісних метрик на навчальній та тестовій вибірці.

Крос-валідація — це один із методів перевірки моделі для оцінки того, як результати статистичного аналізу будуть узагальнені на незалежному наборі даних. Даний метод в основному використовується для оцінки того, наскільки точно модель буде діяти на практиці. У задачі прогнозування моделі зазвичай присвоюється набір відомих вхідних даних, на яких виконується навчання, і набір невідомих даних, з якими тестується модель. Метою крос-валідації є перевірка здатності моделі передбачити нові дані, які не використовувалися при її оцінці, щоб передбачити такі проблеми, як перенавчання, і дати розуміння того, як модель буде узагальнюватися на незалежному наборі даних.

У якості метрик оцінки можуть виступати показники, які характеризують якість моделі з тієї чи іншої сторони. Як наприклад, в даній роботі для порівняння моделей використовуються метрики RMSE, nDCG та Skip. Метрика RMSE – дозволяє оцінити якість прогнозування моделі, в той час як nDCG та Skip дозволяють оцінити якість ранжування результатів пошукової видачі на основі історичних даних.

4.2 Порівняльний аналіз моделей

Для порівняння з розробленою архітектурою моделі в даній роботі були обрані наступні алгоритми: логістична регресія, дерева рішень з наступними методами ансамблювання bagging (RandomForest) та boosting (GradientBoosting).

Процес порівняння проходив наступним чином: для всіх алгоритмів був обраний однаковий навчальний, тестовий, та валідаційний набір вхідних даних. В якості метрик було обрано 3 показники: RMSE, nDCG та Skip.

Також, в якості додаткового критерію були розглянуті графіки розподілу вірогідності вибору/не вибору вакансії. Дана характеристика дозволяє побачити наскільки коректно модель розмежовує ці два випадки. Відповідно, чим більший зсув між двома розподілами, тим краще.

На графіках розподілу вірогідності вибору/не вибору вакансії на осі абсцис відкладається значення вірогідності, на осі ординат щільність розподілу.

На графіках розподілу метрик nDCG та Skip, а також гістограмі метрики RMSE на осі абсцис відображається відповідне значення навчальної вибірки, на осі ординат відкладається відповідне значення метрики.

4.2.1 Модель логістична регресія

На рисунку 4.1 зображений розподіл вірогідності вибору/не вибору вакансій для моделі логістичної регресії, як видно, розподіли розходяться на певному проміжку, але не можна сказати що модель їх повністю розмежовує.

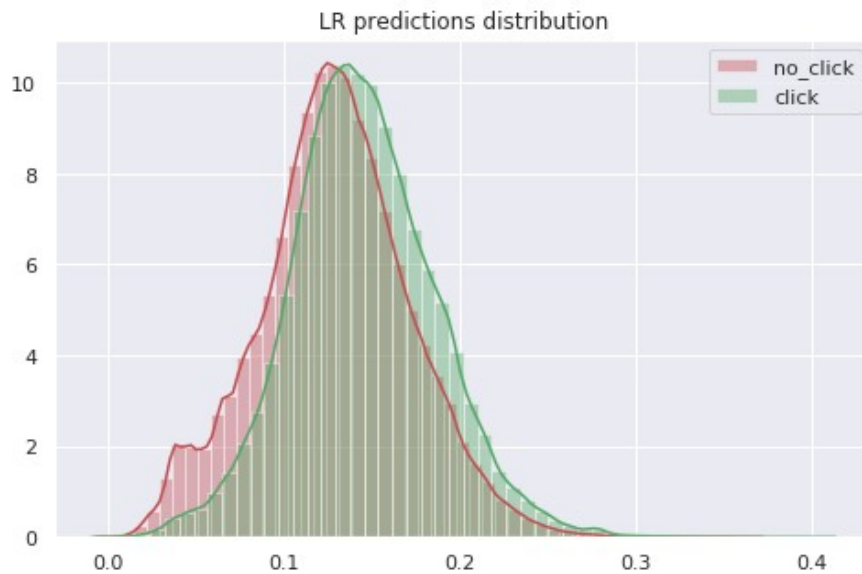


Рисунок 4.1 — Розподіл вірогідності вибору/не вибору вакансії для логістичної регресії

На рисунку 4.2 зображена гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для моделі логістичної регресії, як видно, показники метрики не сильно відрізняються одна від одної, це відповідає тому, що модель не “перенавчилась” на тренувальній вибірці, що загалом непогано.

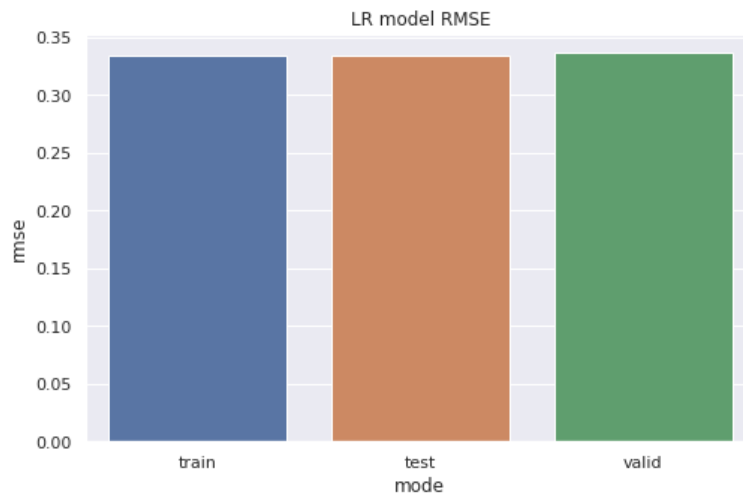


Рисунок 4.2 — Гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для логістичної регресії

На рисунку 4.3 та 4.4. зображено розподіл метрики nDCG та Skip для логістичної регресії, аналогічно як і для метрики RMSE показники не сильно розбігаються, що знову ж таки свідчить про стійкість моделі до “перенавчання”.

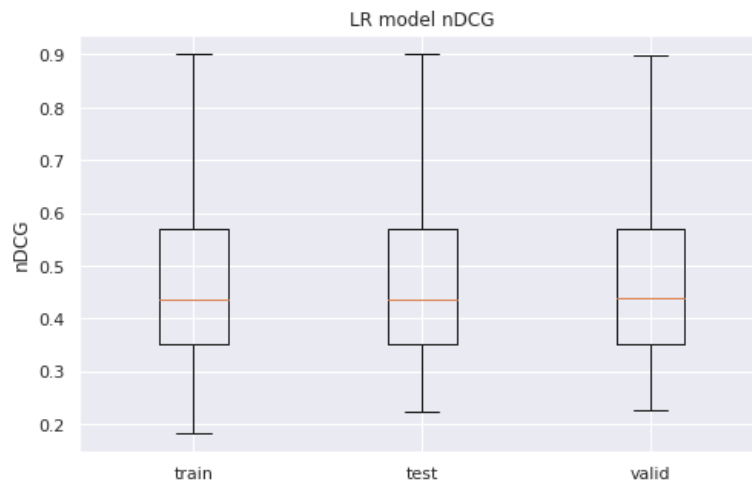


Рисунок 4.3 — Розподіл метрики nDCG на тренувальній, тестовій та валідаційній вибірці для логістичної регресії

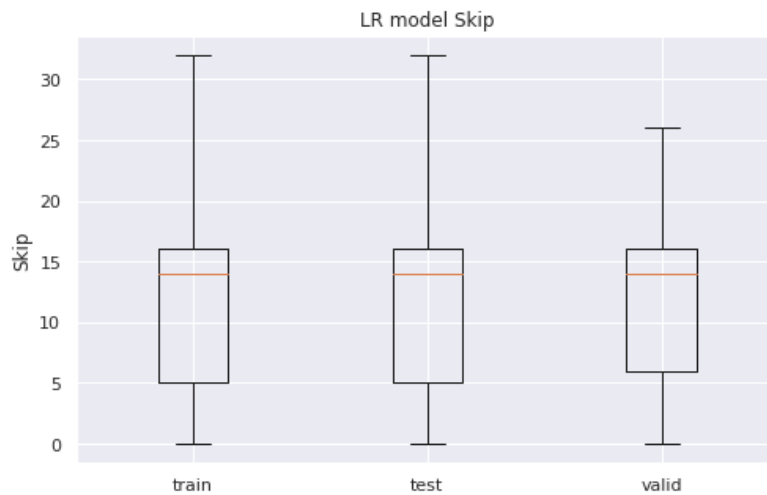


Рисунок 4.4— Розподіл метрики Skip на тренувальній, тестовій та валідаційній вибірці для логістичної регресії

4.2.2 Модель RandomForest

На рисунку 4.5 зображений розподіл вірогідності вибору/не вибору вакансій для моделі RandomForest, на відміну від моделі лінійної регресії, розподіли розходяться, тобто в даному випадку можна сказати що модель їх майже повністю розмежовує.

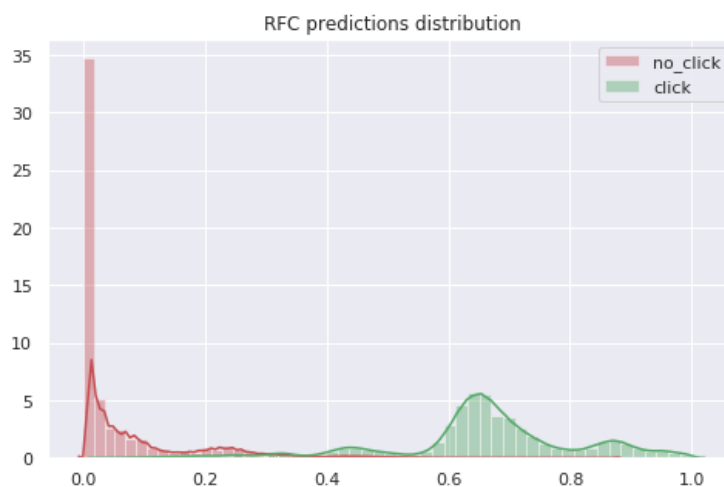


Рисунок 4.5 — Розподілу вірогідності вибору/не вибору вакансії для RandomForest

На рисунку 4.6 зображена гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для моделі RandomForest, як видно, показники метрики сильно просідають на тренувальній вибірці, це відповідає тому, що модель сильно “перенавчилась” на тренувальній вибірці, що свідчить про погані показники генералізації моделі.

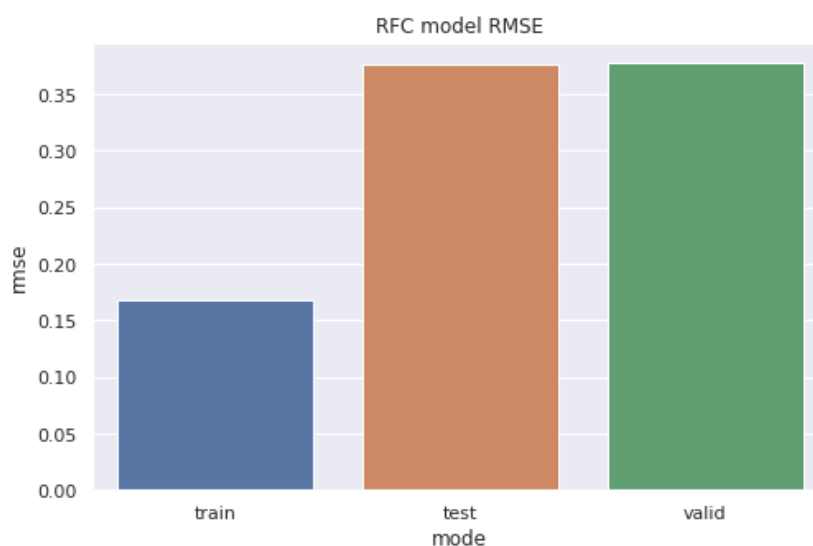


Рисунок 4.6 — Гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для RandomForest

На рисунку 4.7 та 4.8 зображено розподіл метрики nDCG та Skip для моделі RandomForest. На рисунку 4.8 — розподілу метрики Skip, аналогічно як і для метрики RMSE, спостерігається тенденція, яка свідчить про “перенавчання”.

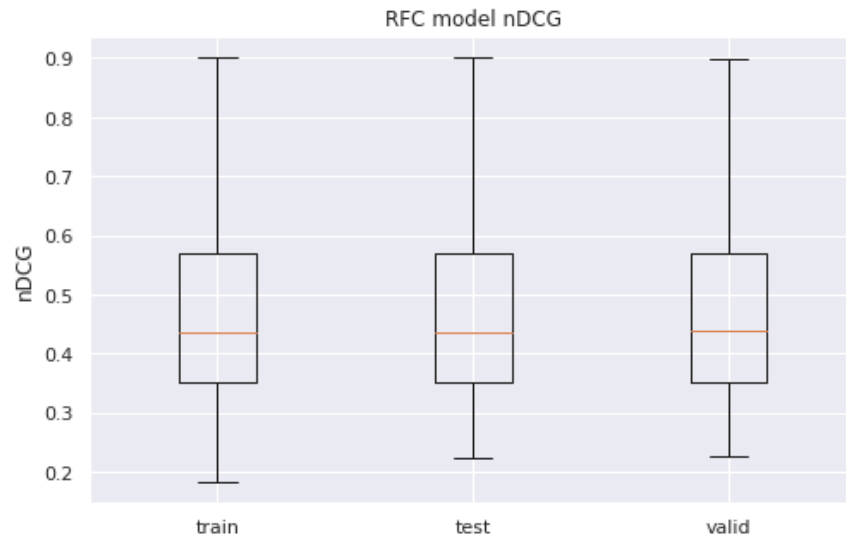


Рисунок 4.7 — Розподіл метрики nDCG на тренувальній, тестовій та валідаційній вибірці для RandomForest

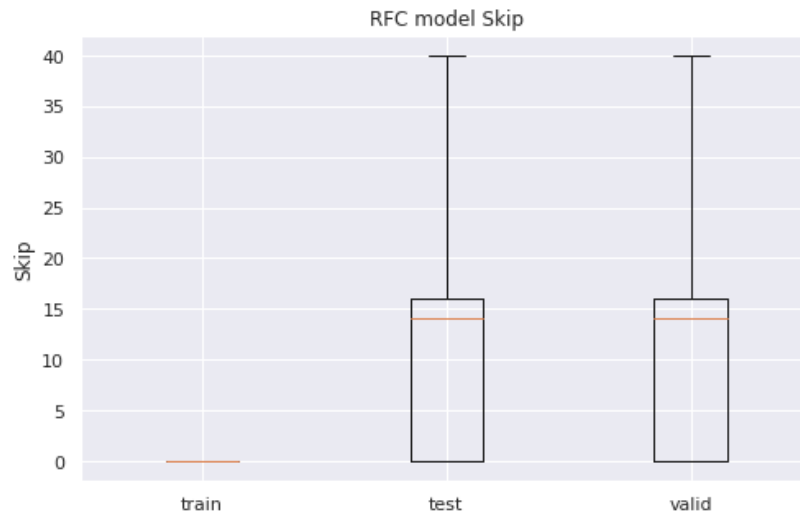


Рисунок 4.8 — Розподіл метрики Skip на тренувальній, тестовій та валідаційній вибірці для RandomForest

4.2.3 Модель GradientBoosting

На рисунку 4.9 зображений розподіл вірогідності вибору/не вибору вакансій для моделі GradientBoosting, як видно, розподіли розходяться на певному проміжку, як і в випадку розподілу для логістичної регресії, але не можна сказати що модель їх повністю розмежовує.

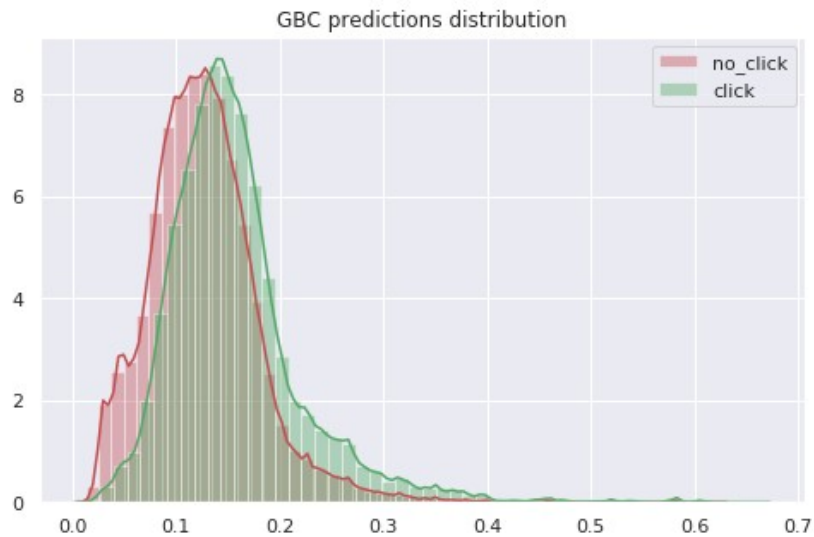


Рисунок 4.9 — Розподіл вірогідності вибору/не вибору вакансії для GradientBoosting

На рисунку 4.10 зображена гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для моделі GradientBoosting, як видно, показники метрики не сильно відрізняються одна від одної, це відповідає тому, що модель не “перенавчилась” на тренувальній вибірці, що загалом непогано. Загалом, показники метрики відповідають аналогічним показникам для моделі логістичної регресії.



Рисунок 4.10 — Гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для GradientBoosting

На рисунку 4.11 та 4.12. зображено розподіл метрики nDCG та Skip для моделі GradientBoosting, аналогічно як і для метрики RMSE показники не сильно розбігаються, що знову ж таки свідчить про стійкість моделі до “перенавчання”. Загалом, показники метрики відповідають аналогічним показникам для моделі логістичної регресії.

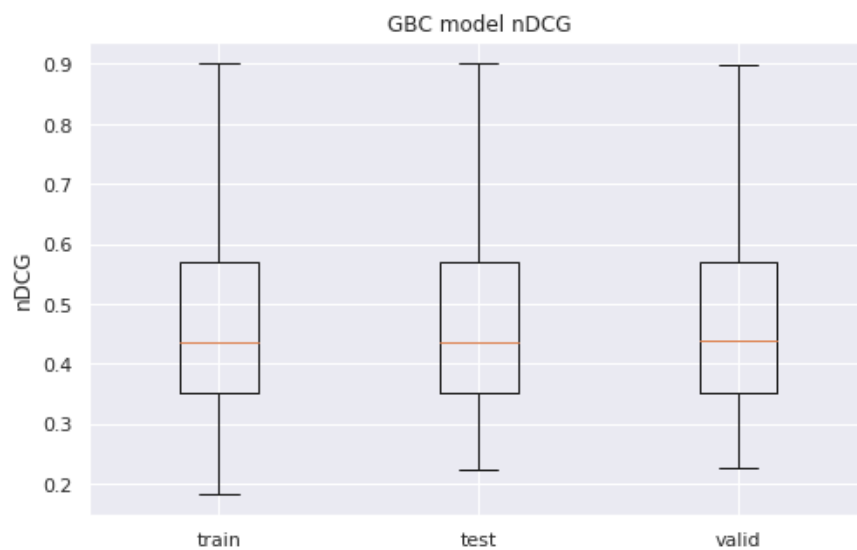


Рисунок 4.11 — Розподіл метрики nDCG на тренувальній, тестовій та валідаційній вибірці для GradientBoosting



Рисунок 4.12 — Розподіл метрики Skip на тренувальній, тестовій та валідаційній вибірці для GradientBoosting

4.2.4 LSTM + Attention Mechanism

На рисунку 4.13 зображений розподіл вірогідності вибору/не вибору вакансій для моделі LSTM + Attention Mechanism, на відміну від попередніх моделей, видно що розподіли трохи розходяться, але в лівому кутку видно характерний стовпчик з найбільшою щільністю розподілу та значенням рівним нулю, це відповідає тому, що модель у більшості випадків чітко розрізняє вірогідність вибору/не вибору вакансії.

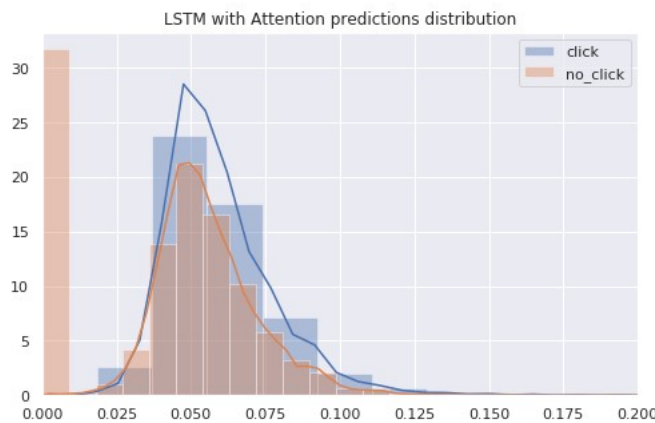


Рисунок 4.13 — Розподіл вірогідності вибору/не вибору вакансії для LSTM + Attention Mechanism

На рисунку 4.14 зображена гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для моделі LSTM + Attention Mechanism, як видно, показники метрики не сильно відрізняються одна від одної, це відповідає тому, що модель не “перенавчилась” на тренувальній вибірці, що загалом непогано. Крім того, показники метрики перевершують показники попередніх моделей приблизно на 21%.

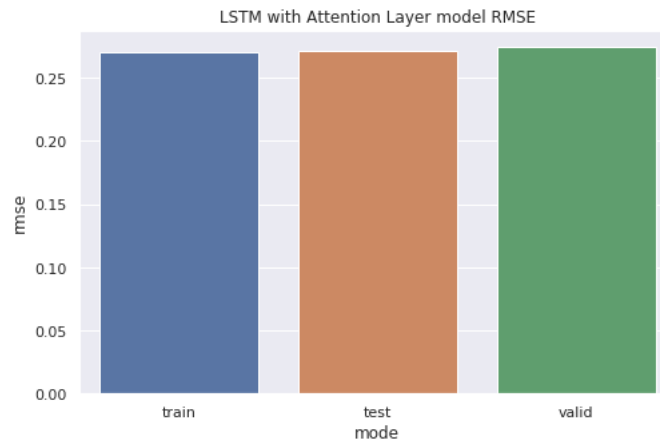


Рисунок 4.14 — Гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для LSTM + Attention Mechanism

На рисунку 4.15 та 4.16. зображено розподіл метрики nDCG та Skip для моделі LSTM + Attention Mechanism, аналогічно як і для метрики RMSE показники не сильно розбігаються, що знову ж таки свідчить про стійкість моделі до “перенавчання”. Загалом, показники метрики трохи відриваються від показників раніше розглянутих моделей.

Незначний, у порівнянні з метрикою RMSE, відрив пояснюється природою розподілу вибору/не вибору вакансій на навчальній вибірці, а саме тим, що в межах одного пошуку в середньому обирається 2 вакансії з 20 запропонованих, і як наслідок кількість не обраних вакансій значно перевищує обрані вакансії, що й спричиняє даний ефект.

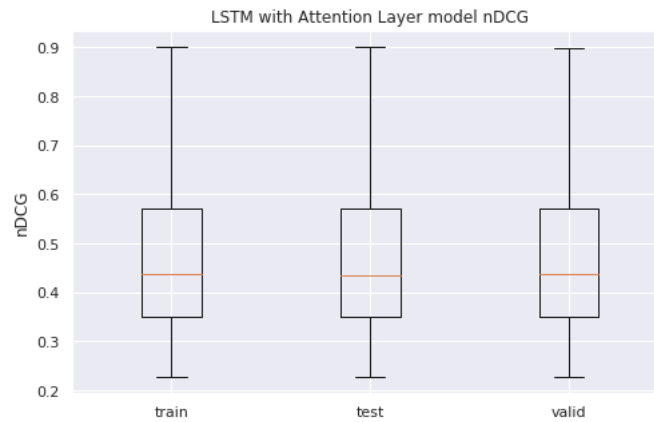


Рисунок 4.15 — Розподіл метрики nDCG на тренувальній, тестовій та валідаційній вибірці для LSTM + Attention Mechanism

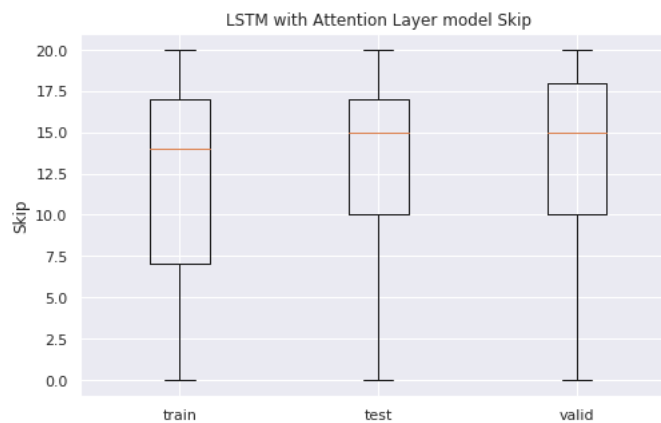


Рисунок 4.16 — Розподіл метрики Skip на тренувальній, тестовій та валідаційній вибірці для LSTM + Attention Mechanism

4.2.5 LSTM + Attention Mechanism + Latent Semantic Model

На рисунку 4.17 зображений розподіл вірогідності вибору/не вибору вакансій для моделі LSTM + Attention Mechanism + Latent Semantic Model, загалом спостерігається аналогічна ситуація, як і в моделі LSTM + Attention Mechanism, однак в лівому кутку знову спостерігається характерний стовпчик з найбільшою

щільністю розподілу, яка майже в два рази більше ніж в попередній моделі, це відповідає тому, що модель більш чітко розрізняє вірогідність вибору/не вибору вакансії.

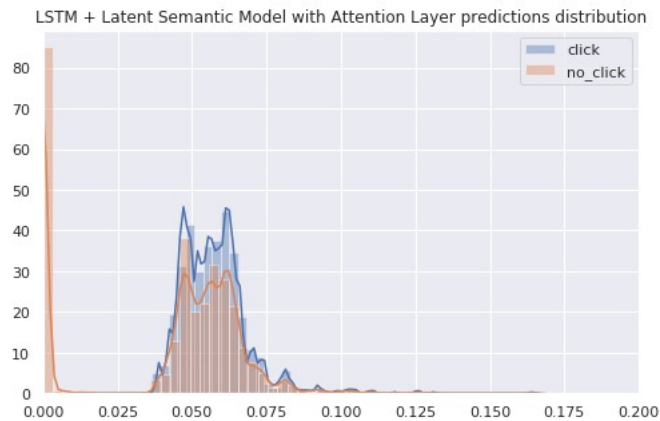


Рисунок 4.17 — Розподіл вірогідності вибору/не вибору вакансії для LSTM + Attention Mechanism + Latent Semantic Model

На рисунку 4.18 зображена гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для моделі LSTM + Attention Mechanism + Latent Semantic Model, як видно, показники метрики аналогічні до показань попередньої моделі.

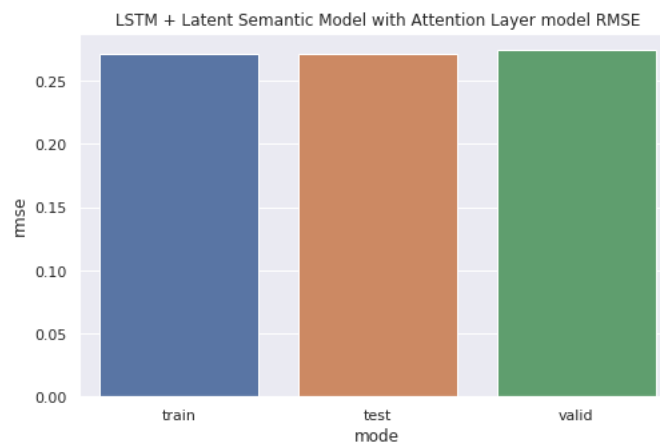


Рисунок 4.18 — Гістограма метрики RMSE на тренувальній, тестовій та валідаційній вибірці для LSTM + Attention Mechanism + Latent Semantic Model

На рисунку 4.19 та 4.20 зображено розподіл метрики nDCG та Skip для моделі LSTM + Attention Mechanism + Latent Semantic Model , знову ж таки, показники майже аналогічні до показників попередньої моделі.

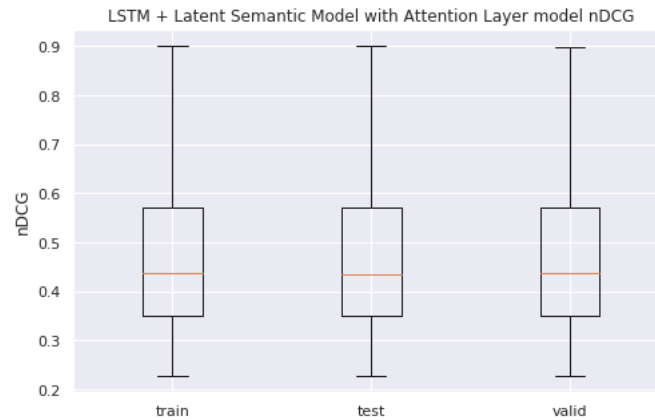


Рисунок 4.19 — Розподіл метрики nDCG на тренувальній, тестовій та валідаційній вибірці для LSTM + Attention Mechanism + Latent Semantic Model

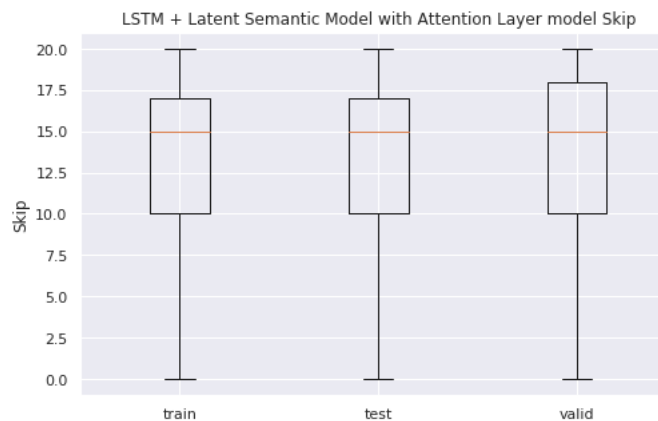


Рисунок 4.20 — Розподіл метрики Skip на тренувальній, тестовій та валідаційній вибірці для LSTM + Attention Mechanism + Latent Semantic Model

4.3 Оцінка генералізації моделі

Оцінка генералізації моделі дозволяє оцінити тенденцію поведінки моделі у реальному світі, тобто на даних, які не використовувалися раніше на тренувальній вибірці. В даному випадку ці дані симулюють тестова та валідаційна вибірки, саме попарна різниця з показниками на тренувальній вибірці дає приблизну оцінку генералізації моделі. Відповідно, чим менша ця різниця, тим модель більш стійка до генералізації.

4.3.1 LSTM + Attention Mechanism

На рисунку 4.21 зображена крива навчання моделі LSTM + Attention Mechanism протягом 120 епох, після кожної епохи розраховувалися метрики RMSE, accuracy, та loss на тестовій та валідаційній вибірці.

На рисунку 4.21 після 80-ї епохи спостерігається тенденція до “перенавчання”, відповідно попарна різниця між показниками на тренувальній та тестовій і валідаційній вибірці починаю рости, що, як згадувалося в розділі 4.3, сприяє погіршенню генералізації моделі. Проте, розрахована різниця дорівнює 0.33% від показників метрик на тренувальній вибірці, що загалом є незначною похибкою.

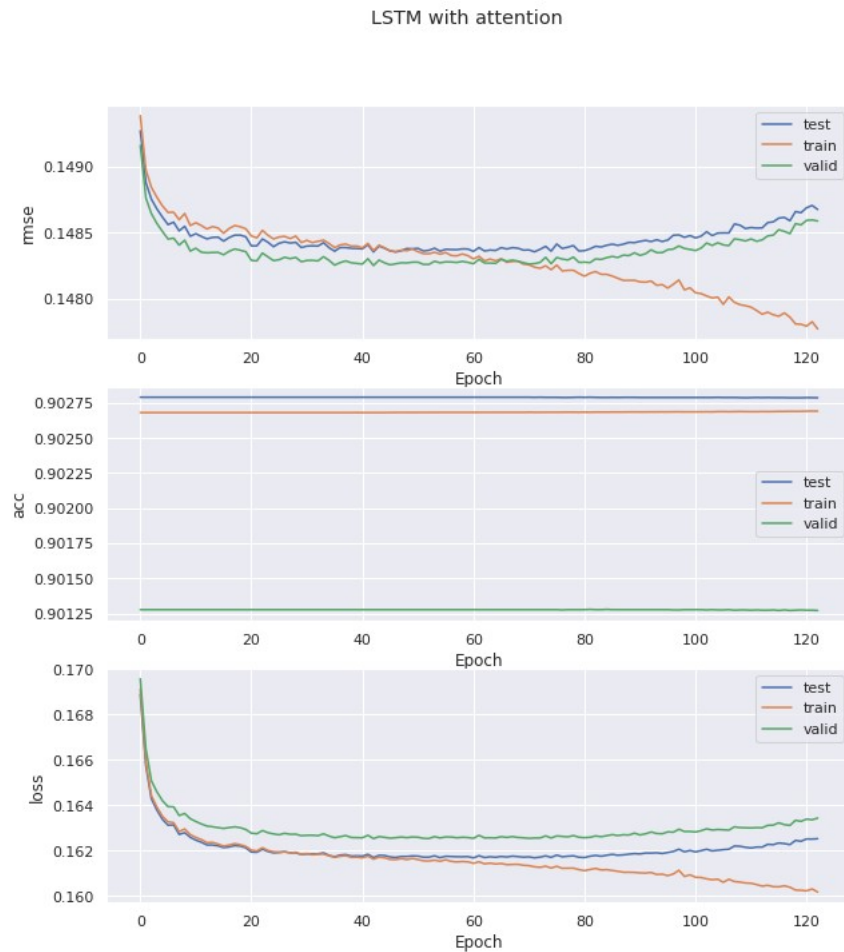


Рисунок 4.21 – Крива процесу навчання моделі LSTM + Attention Mechanism

4.3.2 LSTM + Attention Mechanism + Latent Semantic Model

На рисунку 4.22 зображена крива навчання моделі LSTM + Attention Mechanism + Latent Semantic Model протягом 80 епох, аналогічно, як і для попередньої моделі, після кожної епохи розраховувалися метрики RMSE, accuracy, та loss на тестовій та валідаційній вибірці.

В даній моделі кількість епох дещо відрізняється від попередньої, оскільки після вказаної кількості епох показники метрик на тестовій та валідаційній вибірці не змінювалися, тому навчання нейронної мережі було зупинено.

На відміну від попередньої моделі, попарна різниця між показниками на тренувальній та тестовій і валідаційній вибірці майже відсутня, що свідчить про гарні показники генералізації моделі.

З цього можна зробити висновок, що для вирішення задачі підбору вакансії краще використати модель LSTM + Attention Mechanism + Latent Semantic Model.

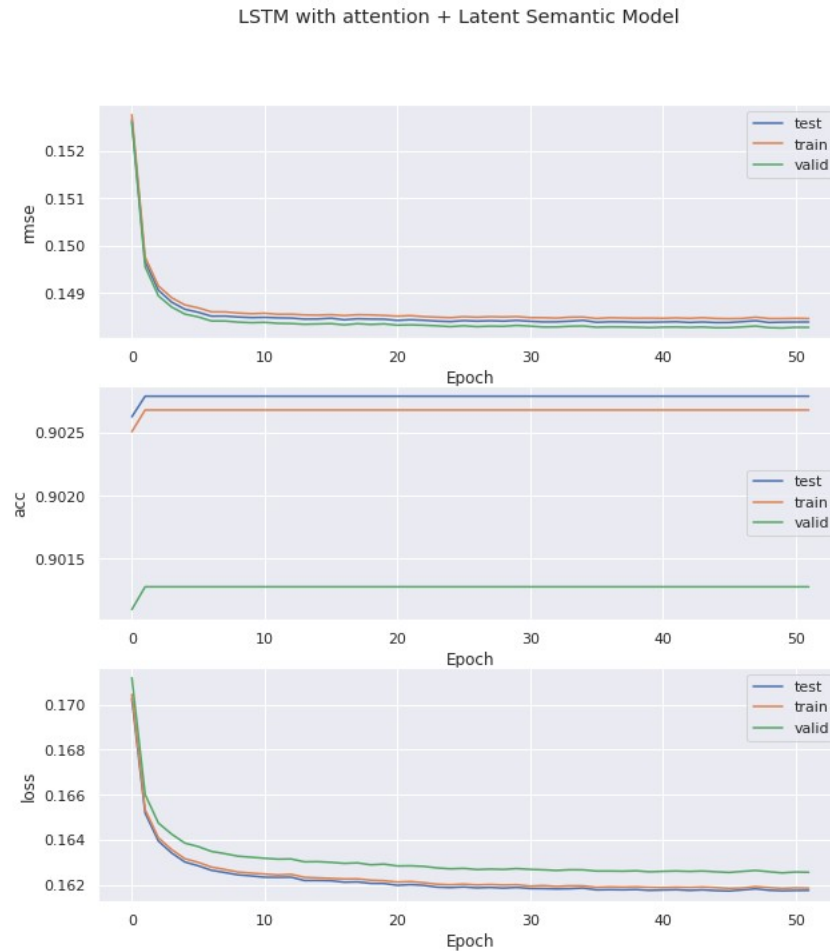


Рисунок 4.22 – Крива процесу навчання моделі LSTM + Attention Mechanism + Latent Semantic Model

4.3.3 Latent Semantic Model

На рисунку 4.23 зображена крива навчання моделі Latent Semantic Model протягом 20 епох, аналогічно, як і для попередньої моделі, після кожної епохи розраховувалися метрики RMSE, ассурау, та loss на тестовій та валідаційній вибірці.

Попарна різниця між показниками RMSE на тренувальній та тестовій і валідаційній вибірці становить 1.3% від показників на тренувальній вибірці, що також свідчить про непогані показники генералізації моделі.



Рисунок 4.23 – Крива процесу навчання моделі Latent Semantic Model

Підсумовуючи, в даному розділі був проведений порівняльний аналіз математичних моделей, які були розроблені та описані в розділі 2. Розроблені математичні моделі порівнювались з базовими алгоритмами машинного навчання.

У ході порівняльного аналізу було визначено, що показники розроблених моделей перевищують показники базових моделей в середньому на 21%, що підтверджує перевагу розроблених математичних моделей.

Також, в роботі був проведений аналіз генералізації розроблених математичних моделей, попарна різниця між показниками на тренувальній та тестовій і валідаційній вибірці для моделі LSTM + Attention Mechanism + Latent Semantic Model майже відсутня, що свідчить про гарні показники генералізації моделі.

З цього був зроблений висновок, що для вирішення задачі підбору вакансії краще використати модель LSTM + Attention Mechanism + Latent Semantic Model.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

В даному розділі викладено маркетинговий аналіз перспектив реалізації системи підбору вакансій на основі історичних даних, як окремого продукту, а також оцінено можливості її ринкового впровадження.

5.1 Опис ідеї проекту

Проект націлений на надання послуг з реалізації та впровадження системи підбору вакансій на основі історичних даних для систем, які спеціалізуються на пошуці роботи. Така система зможе підвищити якість контенту та збільшити конверсію переглядів. Слід зазначити, система може впроваджуватися не лише в системи пошуку роботи, вона може бути використана у будь-якій пошуковій системі, з додатковим визначенням ключових факторів, які властиві сфері впровадження та необхідні для навчання моделі підбору.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система впроваджується в бізнес, для підбору контенту на основі історичних даних, що в свою чергу збільшить конверсію та лояльність	Сайти з пошуку роботи	Збільшення кількості відкликів на вакансії. Збільшення конверсії, покращення показників метрик CTR. Збільшення показників часу перегляду на сайті.

користувачів.

Інформаційні
системи пошуку
контенту.

Збільшення конверсії,
покращення показників
метрик CTR.
Збільшення показників
часу перегляду на сайті.

В даний час на ринку відсутні яскраво виражені конкуренти, що надають послуги та володіють експертизою у сфері з розробки та впровадження систем підбору вакансій, або ж іншого контенту, найчастіше такі рішення розробляються безпосередньо в компаніях або ж замовляються в аутсорсингових ІТ компаніях.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідей проекту

№	Техніко- економіч ні характер истики ідеї	(потенційні) конкурентів	товари/концепції		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
			Мій проект	Сіklum Ерам			
1	Вартість	Середня вартість	Висока вартість	Висока вартість			+

Продовження таблиці 5.2

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна)	S (сильна сторона)
		Мій проект	Ciklum	Eram			
2	Вартість обслуговування	Середня вартість	Висока вартість	Висока вартість			+
3	Прогноз наступної покупки	Так	Ні	Ні			+
4	Зручність інтерфейсу	Зручний інтерфейс	Незручний інтерфейс	Зручний інтерфейс		+	
5	Швидкість впровадження	Швидка	Тривала	Тривала			+

5.2 Технологічний аудит ідеї проекту

Щоб здійснити технологічний аудит ідеї проекту потрібно проаналізувати можливі технології для реалізації продукту.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Back-end частина інтеграції з існуючими системами бізнесу	.Net Core, ASP.NET Core MVC, ADO.NET	Наявні, дороблювати не потрібно.	Доступні, відкритий доступ.
2	Модуль підбору	Python 3.5, Keras,	Наявні, можна доробити\модифікувати	Доступні, відкритий

3	вакансій База даних	NumPy MSSQL Server	та покращити Наявні, дороблювати непотрібно.	доступ. Доступна, за оплату.
---	------------------------	--------------------------	--	------------------------------------

Продовження таблиці 5.3

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	База даних	PostgreSQL	Наявні, дороблювати непотрібно.	Доступна безкоштовна та платна версії.

4

		ClickHouse	Наявні, дороблювати непотрібно.	Доступна безкоштовна та платна версії.
--	--	------------	------------------------------------	--

5	Клієнтський веб інтерфейс	Vue.js	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ.
---	---------------------------------	--------	------------------------------------	-----------------------------------

Обрані технології реалізації ідеї проекту: .Net Core, ASP.NET Core MVC, Python 3.5, Keras, NumPy, MSSQL Server, PostgreSQL, ClickHouse, Vue.js

Для реалізації проекту наявні та доступні усі необхідні технології, що свідчить про технологічну здійсненність проекту.

5.3 Аналіз ринкових можливостей запуску

При проведенні дослідження ринкових можливостей запуску був проведений аналіз основних показників ринку. Аналіз наведений у таблиці 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	Зараз на ринку України відсутні системи підбору вакансій, або ж контенту

2. Загальний обсяг продаж

як окремий продукт, є лише послуги з розробки персональних рішень Український ринок CRM систем за 2017 рік оцінений у 30 млн доларів, що є лише 0.13 частиною світового обсягу [99]. Глобальний ринок CRM показав \$36,8 млрд прибутків за 2017 рік [100]

3. Динаміка ринку (якісна оцінка)

Зростає

4. Наявність обмежень для входу (вказати характер обмежень)

Економія на масштабах, доступ до ресурсів.

Продовження таблиці 5.4

№	Показники стану ринку (найменування)	Характеристика
5.	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	8-30

На даний момент машинне навчання дозволяє підвищити якість сервісів, що в свою чергу покращує показники ключові метрики та KPI бізнесу. За рахунок цього компанії готові вкладати кошти в розробку систем даного типу.

У таблиці 5.5 визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1. Збільшення конверсії;	1. Сайти з пошуку роботи	1. Ціна	1. Якість
2. Підвищення лояльності користувачів;	2. Інформаційні системи пошуку контенту	2. Досвід успішних впроваджень	2. Точність
3. Покращення якості роботи сервісу		3. Репутація постачальника	3. Масштабованість
		4. Функціональність	4. Наявність документації
		5. Відповідність стратегії замовника	5. Технічна підтримка
		6. Вартість впровадження	6. Інноваційність рішення
		7. Вартість підтримки	7. Проста інтеграція
		8. Послуги консалтингу	8. Стабільність роботи

Для проведення аналізу ринкового середовища проведений аналіз факторів, що сприяють ринковому впровадженню (таблиця 5.7) і факторів, що йому перешкоджають (таблиця 5.6).

Негативними факторами при виході на ринок стане несприятливе правове та економічне становище в державі, а також можливість появи нових конкурентів в умовах швидкого розвинення ІТ-індустрії.

Таблиця 5.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1.	Поява нових конкурентів	Можлива поява нових конкурентів через динамічний розвиток ринку ІТ	Цінова та якісна конкуренція.
2.	Економічний	Економічна нестабільність середовища, загальна зниження платоспроможності	Оптимізація бізнес процесів, режим економії вихід на нові зарубіжні ринки.
3.	Правова безпека	Правова незахищеність бізнесу в Україні	Патентування, переведення бізнесу закордон.

До позитивних факторів можна віднести ріст популярності на подібні рішення, сприятливе середовище для розробки подібних рішень.

Таблиця 5.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростаючий ринок впровадження рішень на основі машинного навчання	Масове впровадження сервісів на основі машинного навчання в Україні почалося на проміжку останніх декількох років.	Інтенсивна маркетингова компанія.
2.	Легка доступність людських ресурсів.	Багато ІТ фахівців на ринку	Економія витрат на пошук персоналу.

Для проведення аналізу пропозиції визначаються загальні риси конкуренції (таблиця 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі
---------------------------	---	---

середовища		дії компанії, щоб бути конкурентоспроможною)
Конкурентне середовище ще не до кінця сформоване	Відсутня чітка сегментація ринку рішень на основі машинного навчання	Розвивати нішовий сегмент, щоб захопити найбільшу частку ринку
Національний	Аналізується ринок України	Продаж по всіх регіонах країни
Внутрішньогалузева	Конкуренція лише між компаніями, які розробляють власні системи	Глибокий аналіз ринку та постійний моніторинг конкурентів своєї галузі
Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Характер конкурентних переваг: цінова і нецінова	Цінова. Універсальний характер рішення.	Пошук шляхів зниження собівартості.

Більш детальний аналіз умов конкуренції в обраній галузі економіки проведений за моделлю 5 сил М. Портера. Дані наведені у таблиці 5.9.

Проаналізувавши таблицю 5.9, можна зробити висновок, що можливість виходу на ринок з огляду на конкурентну ситуацію є високою. Для виходу на ринок товар в першу чергу повинен пропонувати унікальні характеристики, які відсутні у продуктах конкурентів. Продукт буде конкурувати за рахунок якості рішення, та простоти в інтеграції з існуючими рішеннями.

На основі аналізу конкуренції, проведеного в таблиці 5.9, а також із урахуванням характеристик ідеї проекту (таблиця 5.2), вимог споживачів до товару

(таблиця 5.5) та факторів маркетингового середовища (таблиці 5.6 та 5.7), визначається та обґрунтовується перелік факторів конкурентоспроможності, що надається у таблиці 5.10.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Прямі конкуренти відсутні. Непрямі конкуренти: компанії, що мають власні системи підбору контенту	Бар'єри для входження на ринок відсутні.	Ринкова сила постачальників дуже слабка	Обмеження платоспроможності	Товарозамінники відсутні
Висновки	Конкурентне середовище ще не до кінця сформоване	Потенційними конкурентами є компанії, що мають власні системи підбору контенту.	Постачальники не диктують умови роботи.	Покупцям легше відмовитися від рішення, ніж платити великі гроші.	Обмеження для роботи на ринку через товари-замінники відсутні.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Універсальний характер застосування	На ринку відсутній готовий продукт універсального характеру застосування.
2.	Ціна	Захоплення ринку за рахунок низької маржинальності продажів та пошуку шляхів зменшення собівартості.
3.	Зручність інтерфейсу	Один з ключових факторів ефективності користування системою.
4.	Масштабованість системи	Один з ключових факторів можливості адаптації системи під зростання бізнесу клієнта.
5.	Якість підбору	Даний фактор безпосередньо впливає на відсоток конверсії.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін

«Інтелектуальна система підбору клієнтського контенту»

№	Фактор конкурентоспроможності	Рейтинг товарів-конкурентів у порівнянні							
		-	-	-	0	+	+	+	+
		-3	-2	-1	0	1	2	3	
1.	Універсальний характер застосування	✓							
2.	Ціна			✓					

Продовження таблиці 5.11

№	Фактор конкурентоспроможності	Рейтинг товарів-конкурентів у порівнянні							
		-	-	-	0	+	+	+	+
		-3	-2	-1	0	1	2	3	
3.	Зручність інтерфейсу				✓				

4. Масштабованість системи
5. Швидкість розробки та впровадження

✓



Фінальним етапом ринкового аналізу можливостей впровадження проекту є проведення SWOT-аналізу (складання матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 5.12). Ринкові загрози та ринкові можливості виступають як наслідки (прогнозовані результати) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (таблиця 5.8, 5.9 аналіз потенційних конкурентів).

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів у таблиці 5.13.

Таблиця 5.12 – SWOT- аналіз стартап-проекту

Сильні сторони:

1. Цінова перевага.
2. Передова технологія, новаторське рішення.

Можливості:

1. Відносно новий ринок для Українського бізнесу – несформоване конкурентне середовище.
2. Багато ІТ фахівців на ринку – економія витрат на пошук персоналу.
3. Зростання попиту на системи

Слабкі сторони:

1. Висока вартість ІТ фахівців.
2. Відсутність експертизи в даній сфері.

Загрози:

1. Можлива поява нових конкурентів через динамічний розвиток рішень на основі машинного навчання.
2. Економічна нестабільність середовища, загальна зниження платоспроможності.
3. Правова незахищеність бізнесу в

підбору контенту.

Україні

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Кооперація Максимізація спільного виграшу	Дуже висока (за рахунок кооперації з існуючими великими гравцями ІТ ринку)	Час реалізації прискорений (1 рік)

Продовження таблиці 5.13

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
2.	Індивідуалізм Максимізація власного виграшу	Середня (адже немає досвіду та репутації для отримання фінансування)	Стандартний час реалізації (1,5-2 роки)
3.	Суперництво Максимізація відносного виграшу	Середня (адже немає досвіду та репутації для отримання фінансування)	Стандартний час реалізації (1,5-2 роки)

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 5.14).

Після визначення та аналізу цільових груп потенційних клієнтів робиться вибір на яких групах потрібно робити фокус, та можна буде будувати стратегію охоплення ринку.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Сайти з пошуку роботи.	Готові	Зростаючий попит (~1 млн долларів)		
2.	Інформаційні системи підбору контенту	Готові	Зростаючий попит (~1 млн долларів)		

Які цільові групи обрано:

Усі, але з пріоритетом на першу групу.

Базові стратегії в обраних сегментах ринку представлені у таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1.	Кооперація	Стратегія концентровано го маркетингу	Цінова перевага. Передова технологія, новаторське рішення.	Стратегія спеціалізації

Залежно від міри сформованості галузевого ринку, характеру конкурентної боротьби, необхідно обрати одну з трьох стратегій конкурентної поведінки: розширення первинного попиту, оборонну або наступальну стратегію або ж застосувати демаркетинг або диверсифікацію (таблиця 4.16).

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першо-прохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1.	Так	Шукати нових	Буде копіювати, але при цьому удосконалювати їх	Стратегія заняття конкурентної ніші

Таблиця 5.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1. Якість	Стратегія спеціалізації	1. Цінова перевага.	1. Підвищення конверсії.
2. Точність		2. Рішення на основі машинного навчання.	2. Підвищення якості роботи сервісу.
3. Масштабованість		3. Легка інтеграція	3. Підвищення показників лояльності користувачі.
4. наявність документації			
5. Технічна підтримка			
6. Інноваційність рішення			
7. Проста інтеграція			
8. Стабільність роботи			

5.5 Розроблення маркетингової програми

Маркетингова програма – це намічений для планомірного здійснення, об'єднаний єдиною метою та залежний від певних строків комплекс взаємопов'язаних завдань і адресних заходів соціального, економічного, науково-технічного, виробничого, організаційного характеру з визначенням ресурсів, що використовуються, а також джерел одержання цих ресурсів [101].

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 5.18 підведені підсумки результатів попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Збільшення конверсії	Аналіз історичних даних для покращення алгоритму підбору контенту	Цінова перевага, легка інтеграція, якість підбору
2.	Підвищення якості роботи сервісу	Безперервне покращення якості підбору контенту	

Продовження таблиці 5.18

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
---	---------	----------------------------	--

3. Ріст
лояльності
користувачів
до сервісу
- Рекомендації на
основі історії
перегляду

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та послуги, його фізичні складові, особливості процесу його надання (таблиця 5.19).

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Товар забезпечує клієнтів механізмом підбору контенту на основі історичних даних для кінцевих користувачів, що збільшує конверсію бізнесу та лояльність кінцевих користувачів

Продовження таблиці 5.19

Рівні товару	Сутність та складові
II. Товар у реальному виконанні	Товар представляє собою систему з набором сервісів, а також інтеграцією в систему замовника.
III. Товар із підкріпленням	До продажу: відбувається презентація можливостей сервісу, та шляхи інтеграції
	Після продажу: відбувається підтримка програмного забезпечення та його покращення

Таблиця 5.20 – Визначення меж встановлення ціни

№ Рівень цін Рівень цін Рівень доходів Верхня та нижня

	на товари- замінники	на товари- аналоги	цільової групи споживачів	межі встановлення ціни на товар/послугу
1.	Від 10000\$	Від 25000\$	> 1млн \$/рік	Нижня межа: 20000\$ за впровадження Верхня межа: 40000\$ за впровадження

Аналіз системи збуту передбачає визначення ефективності кожного елемента цієї системи, оцінювання діяльності апарату працівників збуту. Аналіз витрат обігу передбачає зіставлення фактичних збутових витрат за кожним каналом збуту і видом витрат із запланованими показниками для того, щоб виявити необґрунтовані витрати, ліквідувати затрати, що виникають у процесі руху товарів і підвищити рентабельність наявної системи збуту.

Дані щодо визначення системи збуту надаються в таблиці 4.21.

Таблиця 5.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Лояльність до відомих назв та компаній з високою репутацією, а також тяжіння до низьких цін	Продаж, впровадження і підтримка	Канал нульового рівня	Прямий маркетинг

У якості концепції маркетингових комунікацій були обрані інтегровані маркетингові комунікації. Важливо повністю обдумувати синхронізувати та координувати дії по всіх напрямках каналів комунікацій.

Таблиця 5.22 – Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Особисті знайомства, виступи на тематичних заходах	<div>Медіа</div> <div>Теплі та холодні дзвінки</div> <div>Конференції та тематичні заходи</div>	<div>1.Збільшенн</div> <div>я конверсії.</div> <div>2.Покращенн</div> <div>я якості</div> <div>сервісу.</div> <div>3.Підвищенн</div> <div>я лояльності</div> <div>кінцевого</div> <div>користувача.</div>	Збільшенн я конверсії, дешева та швидка інтеграція	Висвітлит и ефективніс ть рішення та його значущість для розвитку бізнесу та встановле- ння переваги над конкуре- нтами

В цьому розділі було проведено маркетинговий аналіз з метою визначення можливості та доцільності ринкової комерціалізації проекту системи підбору вакансій на основі історичних даних.

Результати дослідження свідчать про можливість ринкової комерціалізації, що обґрунтовується позитивною динамікою нового, ще не до кінця сформованого, ринку, потенціал якого досить значний, судячи з західних більш розвинутих регіонів світу.

При комерційній реалізації проекту можуть стати на заваді економічне та правове становище в країні, проте проект досить легко буде поширювати і за межами України.

При побудові маркетингової компанії варто спиратися на прямий канал збуту нульового рівня, та висвітлювати ефективність рішень такого роду, унікальність рішення та значущість впровадження для встановлення конкурентної переваги, а також на швидке та легке впровадження та інтеграцію.

Підсумок маркетингового аналізу вказує на доцільність подальшої реалізації проекту.

Було проведено маркетинговий аналіз з метою визначення можливості та доцільності ринкової комерціалізації проекту системи підбору вакансій на основі історичних даних.

Результати дослідження свідчать про можливість ринкової комерціалізації, що обґрунтовується позитивною динамікою нового, ще не до кінця сформованого, ринку, потенціал якого досить значний, судячи з західних більш розвинутих регіонів світу.

ВИСНОВКИ

У роботі був проведений огляд існуючих систем пошуку вакансій, а також огляд алгоритмів підбору вакансій. Були визначені недоліки оглянутих алгоритмів. Як наслідок, проаналізувавши недоліки, була розроблена математична модель підбору вакансій, яка дозволяє вирішити вказані проблеми.

Було виконано наступні завдання:

- розглянуто існуючі системи пошук вакансій;
- проаналізовано існуючі алгоритми підбору вакансій;
- побудовано мікросервісну архітектуру системи підбору вакансій;
- розроблено та досліджено математично модель підбору вакансій на основі історичних даних;
- проведено порівняльний аналіз розробленої математичної моделі з існуючими алгоритмами;
- проведено оцінку якості розроблених моделей, і як наслідок, було обрано кінцеву архітектуру математичної моделі.

Була визначена математична модель підбору вакансій, яка включає в себе модель Skip-gram, модель LSTM + Attention Mechanism, а також модель Latent Semantic Model.

Модель Skip-gram дозволяє трансформувати текст вакансії в числовий вектор зі збереження інформації про його контекст, в свою чергу модель Latent Semantic Model дозволяє представити інформацію про пошуковий запит та заголовок вакансії в єдиному семантичному просторі. Модель LSTM + Attention Mechanism представляє собою ядро математичної моделі, яка поєднує результати роботи інших моделей в собі, як наслідок, дозволяє визначити які саме вакансії представити кінцевому користувачу.

Для побудови системи була обрана мікросервісна архітектура, що дозволить у майбутньому легко масштабувати систему, внаслідок чого з'являється можливість легкої інтеграції системи у будь-які інші системи пошуку вакансій.

Крім того, був проведений вибір та обґрунтування елементів та технологій системи пошуку вакансій.

Загалом, система налічує 8 сервісів, які перебувають в безперервній взаємодії між собою. Також, система включає в себе дві основні бази даних, одна для збереження інформації для вакансій, інша — для збереження статистичних даних.

Було проведено успішне тестування системи, а саме, був визначений список відповідних тест-кейсів, у відповідність яким був поставлений список кінцевих умов.

У ході порівняльного аналізу моделей підбору вакансій було визначено, що показники розроблених моделей перевищують показники базових моделей в середньому на 21%, що підтверджує перевагу розроблених математичних моделей.

Внаслідок оцінки якості розроблених моделей було обрано кінцеву архітектуру математичної моделі підбору вакансій на основі показників генералізації моделі, а саме, було обрано модель LSTM + Attention Mechanism + Latent Semantic Model.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

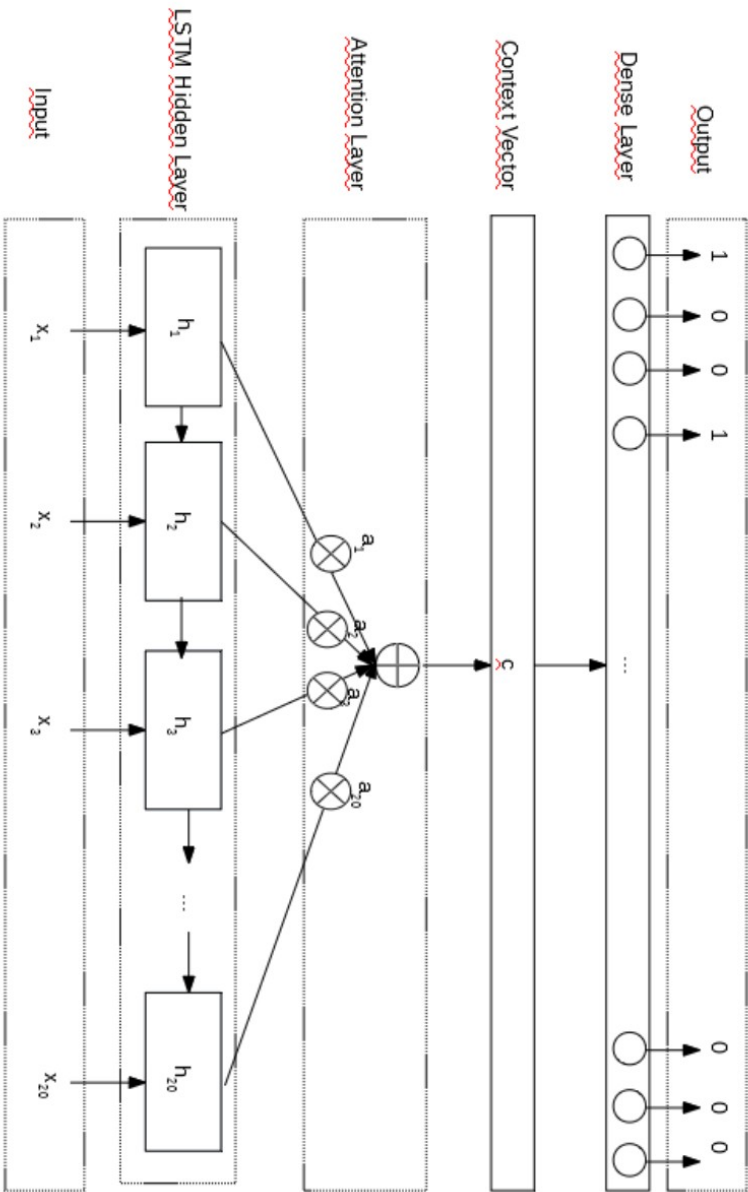
1. Jooble [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Jooble>.
2. About LinkedIn [Електронний ресурс] – Режим доступу до ресурсу: <https://about.linkedin.com/>.
3. Indeed [Електронний ресурс] – Режим доступу до ресурсу: <https://www.indeed.com/about>.
4. Linear Regression [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Linear_regression.
5. Least Squares [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Least_squares
6. Logistic Regression [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Logistic_regression
7. Decesion Tree [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Decision_tree
8. Нейронные сети [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/post/312450/>
9. Classification Accuracy is Not Enough: More Performance Measures You Can Use [Електронний ресурс] – Режим доступу до ресурсу: <https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/>
10. Discounted cumulative gain [Електронний ресурс] – Режим доступу до https://en.wikipedia.org/wiki/Discounted_cumulative_gain
11. Dipanjan S. Implementing Deep Learning Methods and Feature Engineering for Text Data: The Skip-gram Model [Електронний ресурс] / Sarkar Dipanjan – Режим доступу до ресурсу: <https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-skip-gram.html>.

12. Christopher O. Understanding LSTM Networks [Електронний ресурс] / Olah Christopher. – 2015. – Режим доступу до ресурсу: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
13. Colin R. FEED-FORWARD NETWORKS WITH ATTENTION CAN SOLVE SOME LONG-TERM MEMORY PROBLEMS [Електронний ресурс] / R. Colin, E. P. Daniel. – 2016. – Режим доступу до ресурсу: <https://colinraffel.com/publications/iclr2016feed.pdf>.
14. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval [Електронний ресурс] / [S. Yelong, H. Xiaodong, G. Jianfeng та ін.] – Режим доступу до ресурсу: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/cikm2014_cdssm_final.pdf.
15. Microservices [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Microservices>.

ДОДАТКИ

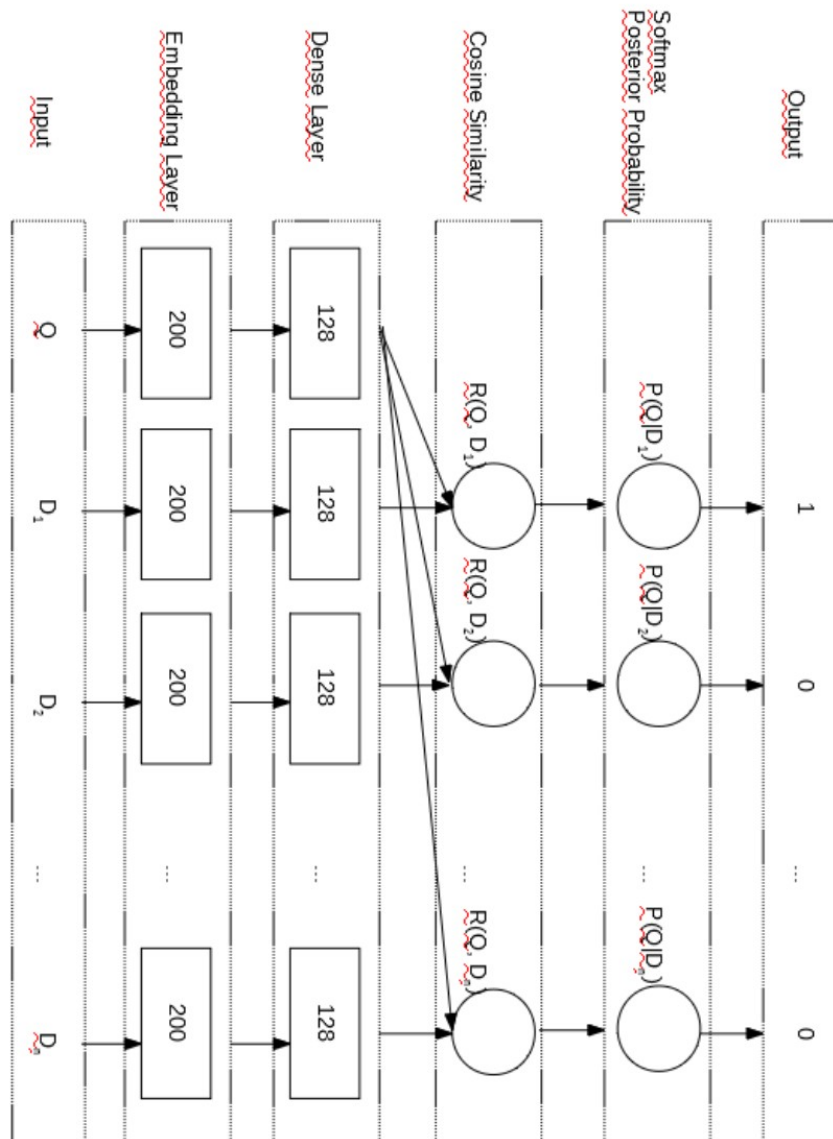
ДОДАТОК А

Структурна схема нейронної мережі LSTM + Attention Mechanism



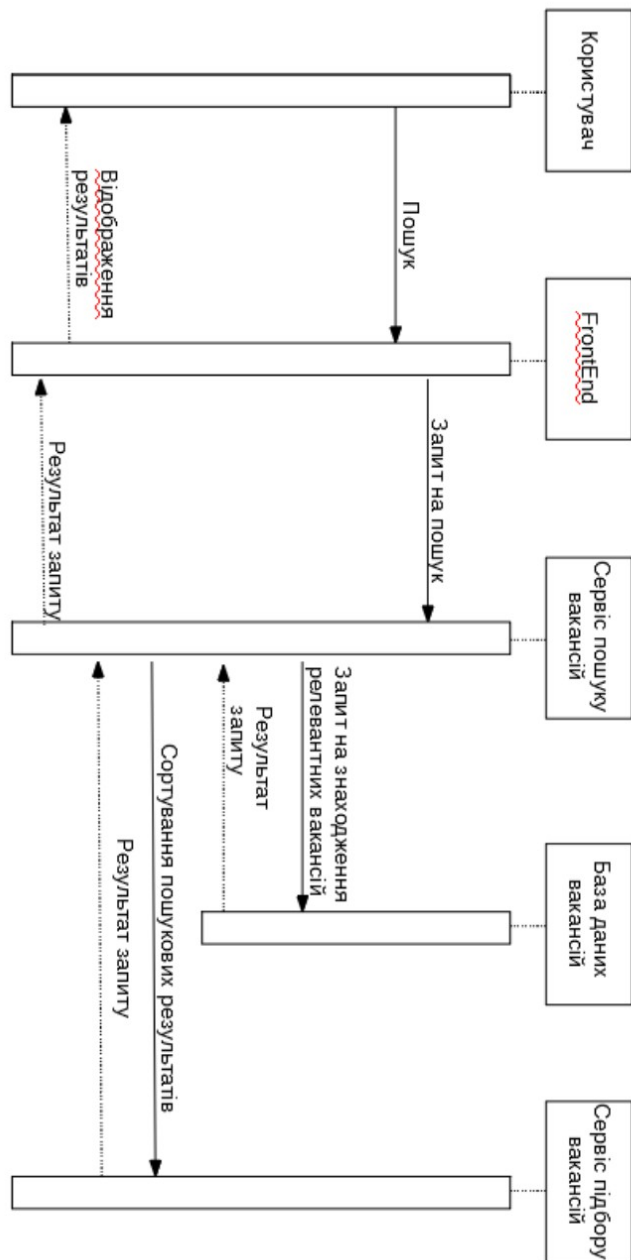
ДОДАТОК Б

Структурна схема нейронної мережі Latent Semantic Model



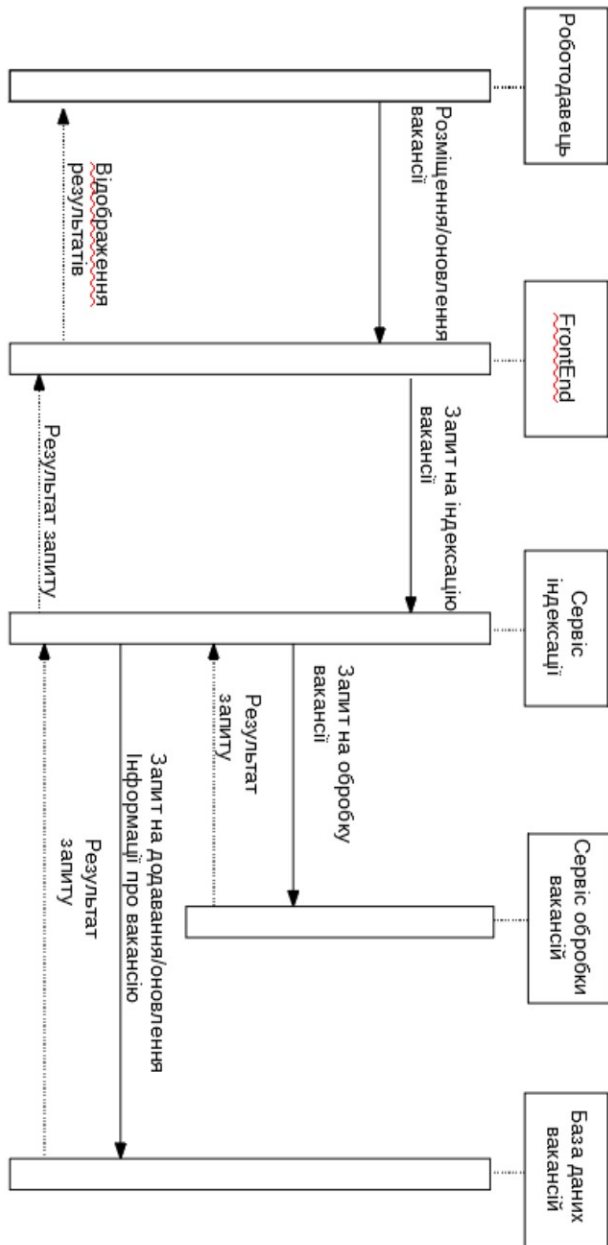
ДОДАТОК В

Діаграма послідовності пошуку вакансій



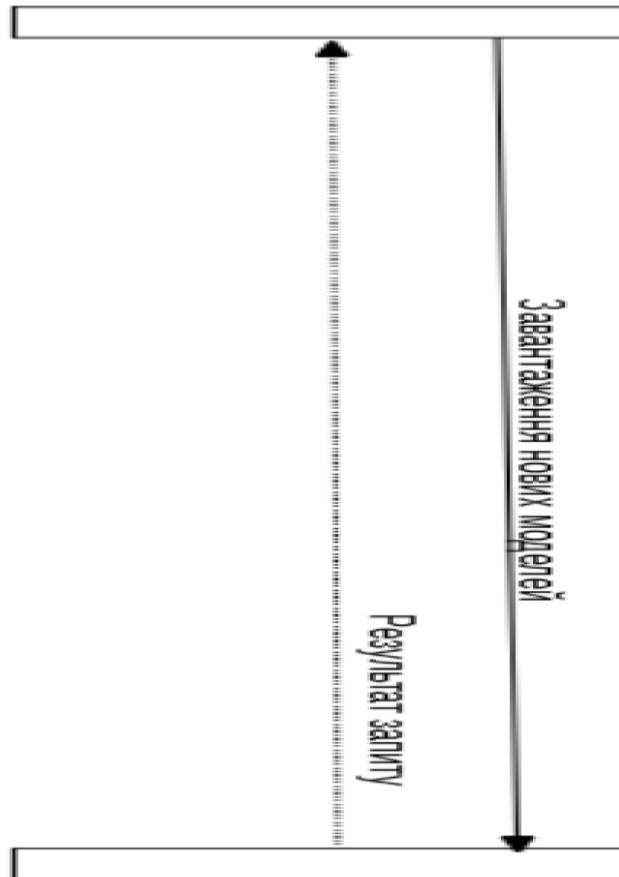
ДОДАТОК Г

Діаграма послідовності розміщення вакансій



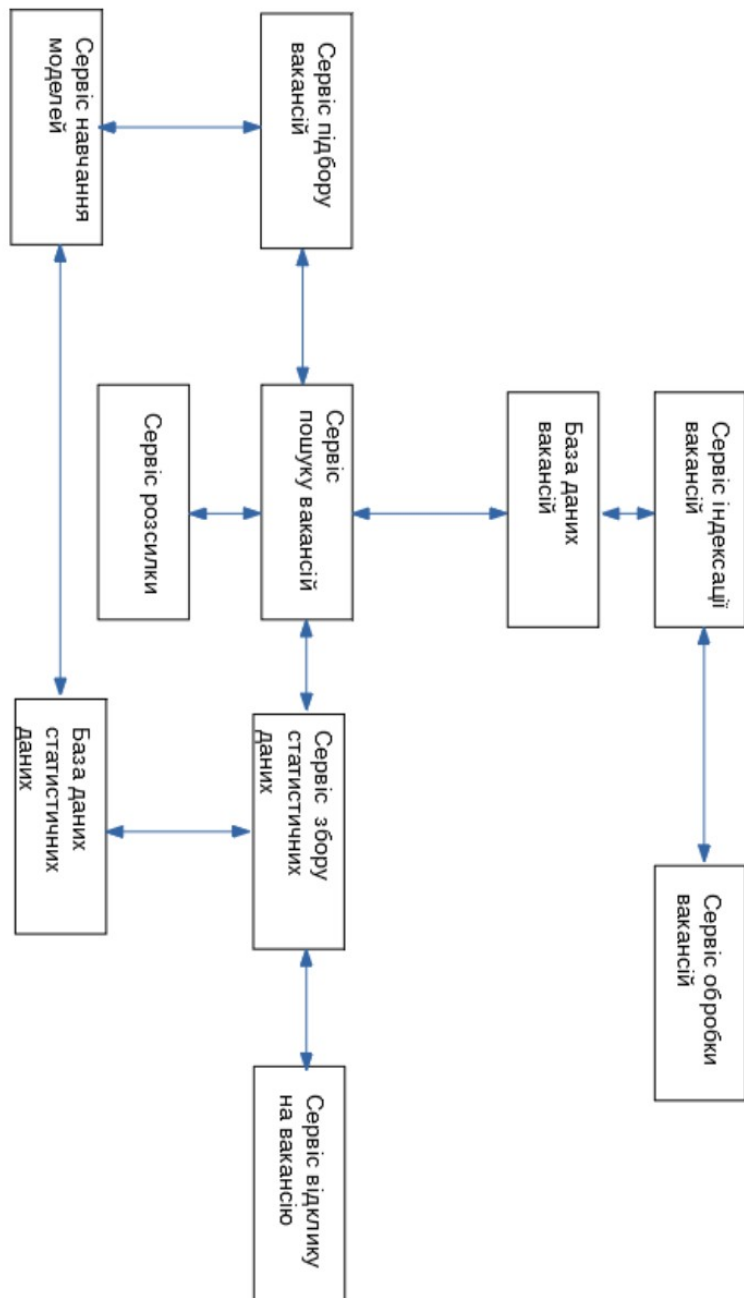
ДОДАТОК Д

Діаграма послідовності навчання моделей



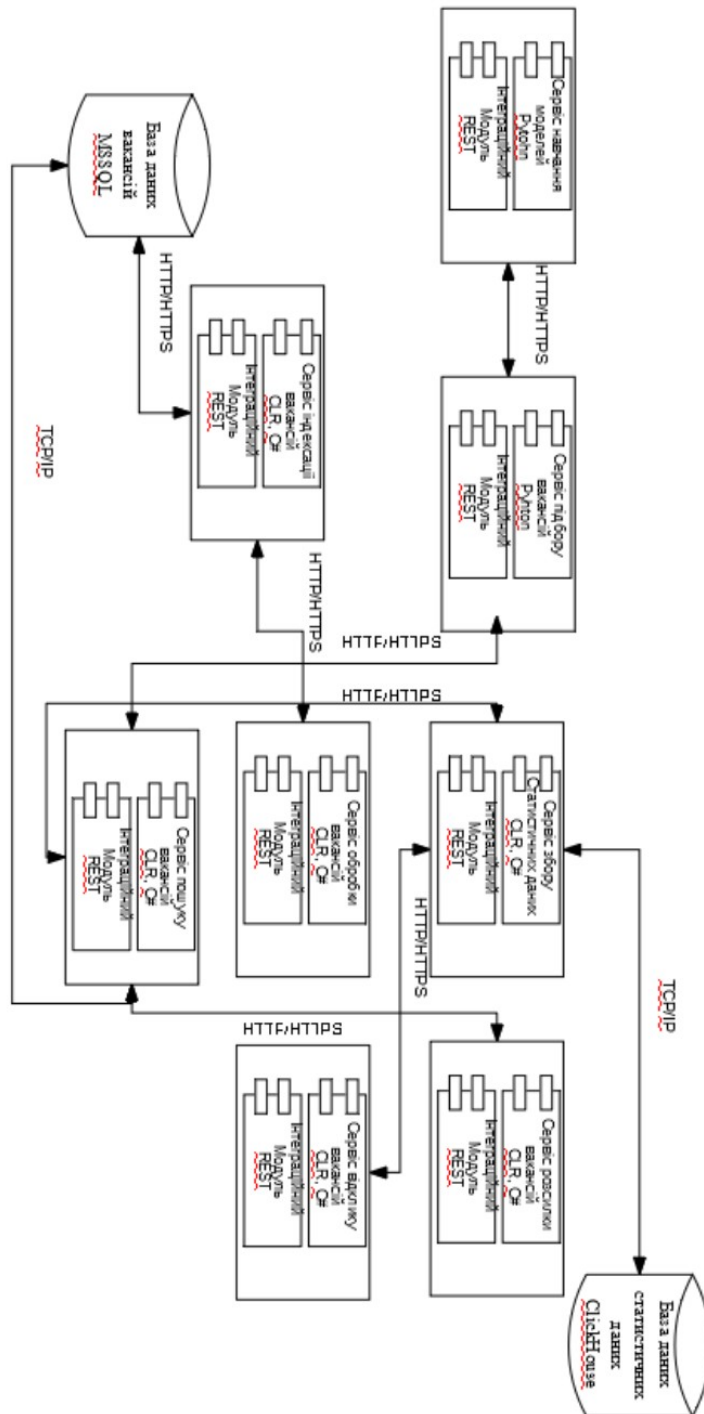
ДОДАТОК Е

Структурна схема системи



ДОДАТОК Ж

Діаграма розгортання



ДОДАТОК И

Діаграма сценаріїв використання

